

**Stefan Künzell**

# **Motorik und Konnektionismus**

**Neuronale Netzwerke als Modell  
interner Bewegungsrepräsentationen**

**Mit einem Vorwort von Klaus Roth**

**bps-Verlag Köln**

**Künzell, Stefan:**

Motorik und Konnektionismus: Neuronale Netzwerke als Modell interner  
Bewegungsrepräsentationen / Stefan Künzell. - Köln: bps-Verl., 1996

(Betrifft: Psychologie & Sport ; 28)

ISBN 3-922386-53-9



1. Auflage 1996

963201

Verlag: bps-Verlag  
Jürgen R. Nitsch und Dieter Hackfort  
c/o Psychologisches Institut  
der Deutschen Sporthochschule Köln

Lektorat: PD Dr. Jörn Munzert

Druck: Druckerei Franz Hansen  
Luxemburger Str. 72  
50674 Köln

Alle Rechte vorbehalten. Ablichtungen und Nachdruck nur mit  
ausdrücklicher Genehmigung des Verlags.

ISBN 3-922386-53-9  
ISSN 0172-5319

## Vorwort

Die Begriffe „Theorie“ und „Unverständlichkeit“ werden in der Diskussion um die (Praxis-)Relevanz sportwissenschaftlicher Forschungsansätze nicht selten als so etwas wie Synonyme interpretiert. Derzeit scheint sich vor allem die *scientific community* der sportbezogenen Motorikforscher um eine Bestätigung dieser Sichtweise zu bemühen. Dem „Theorienovizen“, der an Fragestellungen der motorischen Kontrolle und des motorischen Lernens interessiert ist, werden fortwährend neue, hochkomplexe Modelle dargeboten und als geeignete Nachfolger für die jahrzehntelang dominierenden, heute aber kaum mehr haltbaren Informationsverarbeitungsansätze präsentiert. Die Situation wirkt alles andere als übersichtlich, eben unverständlich; es wird vielfach nicht klar, worin die wesentlichen Unterschiede zwischen den empfohlenen Konzepten bestehen, ob ein Modell wirklich neu ist oder ob nur „alter Wein in neuen Schläuchen“ verkauft wird.

Das von Künzell vorgelegte Buch schwimmt – so betrachtet – gegen den Trend. Erstens verschafft es dem Leser in beispielgebend prägnanter Form einen Überblick über den aktuellen Stand der bewegungswissenschaftlichen Theoriebildung. Es bietet also Strukturierungs- und Einordnungshilfen, die nicht nur für den Laien in diesem Bereich gewinnbringend sein werden. Zweitens „verkauft“ Künzell ausgesprochen „jungen, theoretisch hochprozentigen Wein“. Sein „Anbaugebiet“ ist der Konnektionismus. Dies verwundert nicht, ist Künzell doch ausgebildeter Sportwissenschaftler und Informatiker. Was man sich unter Konnektionismus vorzustellen hat und wie er konkret in die Motorikforschung integriert werden kann, wird am Ende dieses Buches jedem Leser klar geworden sein. Künzell führt auch Nicht-Experten unter Verzicht auf Wortungetüme und kumulierte Formelaufstellungen in die Welt der neuronalen Netze ein. Selbstverständlich und völlig legitim setzt er dabei mehr auf (eine gut begründete) Überzeugungsarbeit als auf Darstellungsneutralität.

Wenn ein Autor durch Überlegungen und Argumentationen überzeugt, ist dies löblich. Noch wichtiger wird für viele allerdings sein, daß Künzell

– im quasi empirischen Teil des Buches – exemplarisch den Nachweis erbringt, daß mit konnektionistischen Ansätzen eine Modellierung (klassischer) empirischer Befunde zum Bewegungslernen prinzipiell möglich ist und daß sie perspektivisch in der Lage sein könnten, zur Lösung offener Probleme der Bewegungskontrolle beizutragen.

Wie bei jedem Buch, kann man auch bei dem Buch von Künzell die Inhalte in zwei Teile gruppieren (vgl. Heuer 1983, 7). Den einen findet der Leser einleuchtend, den anderen nicht. Der Urheber des ersten Teils ist ausschließlich der Autor selbst; beim zweiten Teil kommt möglicherweise der Einfluß der Gruppe der Berliner Bewegungswissenschaftler zum Tragen. Stefan Künzell hat in den Jahren 1990-1995 maßgeblich die in diesem Kreis geplanten und realisierten Forschungsarbeiten mitgestaltet.

Heidelberg, im September 1995

Klaus Roth



## Danksagung

Ideen für Themen wissenschaftlicher Arbeiten entstehen zumeist nicht im Kopf des Autors allein, sondern beruhen oft auf Denkanstößen aus sehr unterschiedlichen Quellen. Ein wichtiger Denkanstoß für mich war die Teilnahme an der exzellenten Vorlesung „Einführung in die Theorie der neuronalen Netze“ von Raúl Rojas im Fachbereich Mathematik und Informatik der FU Berlin. Dieser Vorlesung nebst dem ausführlichen Vorlesungsskript, das inzwischen als Buch vorliegt (Rojas 1993) und einem anschließenden Hauptseminar verdanke ich einen wesentlichen Teil meiner Kenntnisse über neuronale Netze.

Ermutigt zu der Übertragung dieser Kenntnisse auf sportwissenschaftliche Zusammenhänge haben mich die Mitarbeiterinnen und Mitarbeiter der Arbeitsgruppe Bewegungswissenschaft am Institut für Sportwissenschaft der FU Berlin, vor allem Ernst-Joachim Hossner. Durch die vielfältigen Diskussionen in den Colloquien, aber auch während der beiden Berliner Winterakademien, an denen ich teilnehmen durfte, lernte ich, wie spannend und dynamisch die Entwicklung der Theorien zum motorischen Lernen und zur motorischen Kontrolle sind. In diesen Gesprächen entstand – im Gegensatz zu der im Vorwort geäußerten Vermutung (und hier muß ich meinem Lehrer, Klaus Roth, erstmals in einer Publikation widersprechen) – vor allem der einleuchtende Teil dieses Buches. So ist diese Arbeit eine glückliche Chance, aber auch eine schwierige Herausforderung, die drei Wissenschaftsdisziplinen, zu denen ich mich hingezogen fühle, nämlich Sportwissenschaft, Informatik und Psychologie miteinander zu verbinden.

Bedanken möchte ich mich auch bei Jörn Munzert, der mich erst auf die Idee brachte, die vorliegende Arbeit als Buch zu veröffentlichen, bei Sabine Reuker, die sich mit dem Ausmerzen der Rechtschreibfehler abmühte, und bei Doris Schuhmann, die sich während der Schlußphase meiner Examensarbeit, die die Grundlage für das vorliegende Buch bildet, um mein leibliches und seelisches Wohl kümmerte.

Mein besonderer Dank gilt meinen Eltern, die liebevoll – eher fördernd als fordernd – mir einen nichtlinearen Lebenslauf ermöglichten, der vielleicht eine Voraussetzung für mein Interesse und Verständnis für nichtlineare Beziehungen in der Bewegungsproduktion und anderswo ist.

Berlin, im September 1995

Stefan Künzell

# Inhaltsverzeichnis

Einleitung .....	11
<b>1 Zum Stand bewegungswissenschaftlicher Forschung .....</b>	<b>13</b>
1.1 Modellierung auf der komputationellen Ebene: Der <i>action approach</i> .....	15
1.1.1 Bernsteins Synergien .....	17
1.1.2 Psychoökologie .....	18
1.1.3 Synergetik .....	19
1.2 Modellierung auf der algorithmischen Ebene: Der <i>motor approach</i> .....	21
1.3 Modellierung auf der implementationellen Ebene .....	23
1.3.1 Optimierungsverfahren .....	25
1.3.2 Die <i>mass-spring</i> -Hypothese .....	26
1.4 Stärken und Schwächen der verschiedenen Herangehensweisen .....	28
1.5 Warum eine neue Theorie? .....	31
<b>2 Was ist Konnektionismus? .....</b>	<b>34</b>
2.1 Definitionen und Terminologie .....	34
2.2 Historische Wurzeln .....	35
2.3 Architektur künstlicher neuronaler Netze .....	39
2.4 Lokale Informationsverarbeitung .....	41
2.5 Globale Informationsverarbeitung .....	45
2.6 Lernen .....	46
2.6.1 Überwachtes Lernen .....	46
2.6.2 Unüberwachtes Lernen .....	48
2.7 Interne Repräsentationen .....	48
<b>3 Konnektionistische Modelle motorischer Kontrolle und     motorischen Lernens .....</b>	<b>52</b>
3.1 Das VITE-Modell von Bullock und Grossberg .....	52
3.1.1 VITE in der wissenschaftlichen Diskussion .....	57
3.1.2 Das Problem der Freiheitsgrade .....	60
3.1.3 FLETE und AVITE .....	61
3.1.4 Fazit .....	62

3.2 Jordans Netz .....	63
3.2.1 Architektur .....	64
3.2.2 Das Problem der Freiheitsgrade .....	69
3.2.3 Das dynamische Modell .....	70
3.2.4 Einordnung in den sportwissenschaftlichen Zusammenhang .....	72
3.3 Vergleich von Jordans Modell und VITE .....	75
3.4 Andere Modelle .....	76
3.4.1 Das M-Net .....	76
3.4.2 Das Kaskaden-Netz .....	78
<b>4 Simulation .....</b>	<b>80</b>
4.1 Grundsätze .....	80
4.2 Aufgabe .....	83
4.3 Struktur des Programms.....	91
4.4 Das Training des Bewegungsvorstellungsnetzes .....	93
4.5 Das Training des Bewegungssteuerungsnetzes .....	98
4.6 Kontextinterferenz .....	99
4.6.1 Hypothesen .....	100
4.6.2 Design .....	100
4.6.3 Betrachtung einzelner Lernverläufe .....	101
4.6.4 Deskriptive Auswertung .....	106
4.6.5 Varianzanalyse .....	110
4.6.6 Interpretation .....	112
4.7 Freiheitsgrade .....	116
4.8 Einordnung und Bewertung .....	123
<b>5 Perspektiven konnektionistischer Bewegungsforschung.....</b>	<b>125</b>
5.1 Bedeutung .....	125
5.2 Stärken .....	128
5.3 Schwächen .....	131
5.4 Sportpraktische Auswirkungen.....	134
5.5 Perspektiven .....	135
Literaturverzeichnis .....	136
Anhang: Quellcode des erstellten Programms.....	144
Personenregister .....	153

## Einleitung

These are revolutionary times for the brain and cognitive sciences. At the heart of the revolution is connectionism. Neural networks - systems consisting of simple neuronlike computing elements - have become a subject of intense interest to scientists spanning a broad range of disciplines including mathematics, physics, biology and psychology (Hanson & Olson 1990, 1).

In der Kognitionsforschung der heutigen Zeit zeichnet sich ein Paradig-  
mawechsel ab. Während vor 10 Jahren noch der herkömmliche „von-  
Neumann“-Computer, eine Maschine zur sequentiellen Datenverarbei-  
tung, das in der Psychologie bevorzugte Modell zur Beschreibung kogni-  
tiver Prozesse war, wird heutzutage die Funktion der Neuronen im Hirn  
selbst zur Modellbildung herangezogen. Es wurde entdeckt, daß es  
möglich ist, durch ihr paralleles und verteiltes Zusammenwirken Infor-  
mation zu speichern und zu verarbeiten, ohne daß eine zentrale Instanz  
explizite Berechnungen oder Zuweisungen durchführt. Im Vordergrund  
des Erkenntnisprozesses steht dabei nicht die Funktion des einzelnen  
Neurons, sondern die sich aus den vielen flexiblen Verbindungen (engl.  
*connections*) zwischen ihnen ergebenden Interaktionen. Daher rührt  
auch der Name „Konnektionismus“, der sich mehr und mehr als Begriff  
für das neue Paradigma in der Psychologie durchsetzt. Inzwischen hat  
sich diese Art der Modellbildung so etabliert, daß sich ein eigener Wis-  
senschaftszweig, die „Neurowissenschaften“, gebildet hat, in der  
Psychologie, Informatik, Neurologie und andere Wissenschaften inter-  
disziplinär zusammenarbeiten. Dies wird auch auf der Hochschulebene  
durch Institutsgründungen, so z.B. der des Instituts für Neuroinformatik  
an der Ruhr-Universität Bochum, bestätigt. Der überwiegende Teil der  
Forschungsarbeit im Bereich der Neurowissenschaften beschäftigt sich  
mit Phänomenen wie (vor allem visuellem und sprachlichem) Wahrneh-  
men und Wiedererkennen. Jedoch gibt es auch – hauptsächlich im US-  
amerikanischen Raum – Arbeiten zur Motorikforschung vor dem Hinter-  
grund des konnektionistischen Paradigmas, die als Grundlage sportwis-  
senschaftlichen Denkens und Forschens durchaus geeignet wären. So

schreiben auch Stelmach und Requin im Vorwort des von ihnen herausgegebenen Buchs:

Since the late 1970s, much has changed in the motor control and learning area. Today, the alliance with Experimental Psychology has diminished and in its place is an emerging federation with Cognitive-Neuroscience. (...) Closed- and open-loop control notions have been replaced by distributed control theories, and mass springs by oscillators and memory representations by neural networks (1992, V).

Die daraus entstandenen Modelle können gut als Alternative zu, vielleicht sogar als Bindeglied zwischen den herkömmlichen Ansätzen dienen. In der deutschen Sportwissenschaft ist dieser Forschungszweig jedoch noch nicht entwickelt. Dies mag zum einen daran liegen, daß die bisherigen Arbeiten in diesem Bereich sich noch sehr mit den Grundlagen motorischen Verhaltens beschäftigen und wohl kaum Aussagen zu Fragestellungen der Sportpraxis zulassen, zum zweiten, daß sich die zentralen Aussagen durch den hohen mathematischen Abstraktionsgrad, der zu ihrer Beschreibung nötig ist, nicht unmittelbar erschließen lassen.

Die vorliegende Arbeit soll einen Beitrag dazu leisten, die Diskussion in der deutschen Sportwissenschaft über einen konnektionistischen Ansatz in der Motorikforschung anzuregen. Es wird überprüft, ob er in der Lage ist, empirische Befunde des menschlichen Bewegungslernens zu modellieren und offene Probleme der Bewegungskontrolle zu lösen.

Nachdem in einem kurzen Überblick in Kapitel 1 die zur Zeit aktuellen Ansätze der Motorikforschung kritisch beleuchtet und in Kapitel 2 die grundlegenden Ideen des Konnektionismus dargelegt worden sind, werden im zentralen Kapitel 3 Konnektionismus und Motorik zusammengedacht und die bisher entwickelten neuronalen Netze als Modelle interner Bewegungsrepräsentationen vorgestellt.

Um einen Einblick in die Vorgehensweise konnektionistischer Modellbildung zu geben und gleichzeitig ihre Angemessenheit und Nützlichkeit zu demonstrieren, wird in Kapitel 4 die Durchführung einer Simulation mit einem selbst implementierten neuronalen Netz dokumentiert. Kapitel 5 rundet das Thema mit dem Versuch eines Ausblicks über die weitere Bedeutung konnektionistischer Motorikforschung ab.

# 1 Zum Stand bewegungswissenschaftlicher Forschung

In der Bewegungswissenschaft wird seit einiger Zeit eine Kontroverse über die richtige Herangehensweise an das Problem der Beschreibung und Erklärung von Kontrolle und Erlernen menschlicher Willkürbewegungen geführt. Spätestens seit dem „*Workshop on Complex Motor Behaviour – Control and Learning in Sports and Everyday Life*“ vom 5.-8. November 1985 am Zentrum für interdisziplinäre Forschung in Bielefeld firmiert sie unter dem Namen „*the motor-action controversy*“. Hauptunterschied zwischen beiden Theorierichtungen ist die Sichtweise der Bewegungsproduktion.

Die Vertreter des *motor approach* lehnen sich an das in der Psychologie vorherrschende Informationsverarbeitungsparadigma an. Analog zu einem Computer, der mit Programmen Befehle erzeugt, welche die peripheren Geräte steuern, generieren nach dieser Ansicht im zentralen Nervensystem (ZNS) gespeicherte Programme Muskelbefehle, welche die Bewegung der peripheren Körperteile bestimmen.

Die Befürworter des *action approach* bezweifeln, daß Bewegungen zentral gesteuert werden, wobei das „Problem der Freiheitsgrade“, auch „Bernsteins Problem“ genannt, eine zentrale Rolle spielt. Statt dessen nehmen sie an, daß Bewegungen sich aus nichtlinearen physiologischen, biomechanischen oder umgebungsbedingten dynamischen Randbedingungen (*constraints*) quasi von allein ergeben.

Knapp zehn Jahre nach der Bielefelder Konferenz ist die Kontroverse noch immer aktuell. Die Kritik am *motor approach* hat sich ausdifferenziert, teilweise wird versucht, Konzepte des anderen Lagers in der eigenen Theorie zu verarbeiten, aber keine der beiden Herangehensweisen kann von sich behaupten, als Siegerin aus der Auseinandersetzung hervorgegangen zu sein, erst recht konnte bis heute kein „kritisches Experiment“ konstruiert werden. Daher liegt der (auch schon früher geäußerte, vgl. Schmidt 1988a, 13) Verdacht nahe, daß sich den Problemen der menschlichen Bewegung auf verschiedenen Ebenen genähert wird. Diese Ansicht soll hier unter Zugrundelegung des Ebenenmodells von

Marr (1982, 24ff) untermauert werden. Marrs Modell hat sich in der Psychologie zur Strukturierung kognitiver Prozesse bewährt. Es wurde ursprünglich für informationsverarbeitende Maschinen entwickelt. Marr unterscheidet drei Ebenen, auf denen diese verstanden werden müssen: Die komputationelle, die algorithmische und die implementationelle Ebene. Auf der *komputationellen Ebene* geht es um eine formale Analyse eines häufig in Alltagssprache formulierten Problems. Dadurch wird es erst einer Lösung durch informationsverarbeitende Maschinen zugänglich gemacht. Auf der *algorithmischen Ebene* werden Fragen nach der Effizienz, der Repräsentationsform, der Fehleranfälligkeit gegenüber zufälligen Eingabeschwankungen usw. gestellt. Auf der *Implementationsebene* geht es darum, die tatsächliche physikalische Realisation der Repräsentationen und des Algorithmus' festzulegen.

Übersetzt in die Sprache der Motorik interessieren auf der komputationellen Ebene Fragen nach dem Ziel der Bewegung und den sie beeinflussenden Randbedingungen; auf der untersten, der implementationellen Ebene, geht es um den konkreten Inhalt der Steuerungssignale an die Muskulatur, den tatsächlichen Bewegungsverlauf und die darin enthaltenen Gesetzmäßigkeiten. Wie die Steuerungssignale nach der Auswahl der Bewegung erzeugt werden, ob und wie Bewegungsproduktionen zentral verarbeitet werden, ob und wie Sich-Bewegen gelernt werden kann ist Gegenstand von Überlegungen auf der algorithmischen Ebene.

Hier soll im folgenden gezeigt werden, daß der *action approach* der komputationelle Ebene zuzuordnen ist, während der *motor approach* sich mit Problemen der algorithmischen Ebene beschäftigt. Einige weitere Modelle, die weder dem einen noch dem anderen Lager zugeordnet werden können, werden als der implementationellen Ebene zugehörig identifiziert. Dabei soll nicht verkannt werden, daß es (wie auch in Marrs Modell) zu Überschneidungen zwischen den Ebenen kommen kann. Die Zuordnung der verschiedenen Herangehensweisen zu den einzelnen Ebenen soll also nicht heißen, daß Aussagen zu anderen Ebenen prinzipiell nicht möglich wären, sondern deutlich machen, wo der Schwerpunkt des Erkenntnisinteresses liegt.



## 1.1 Modellierung auf der komputationellen Ebene: Der *action approach*

Bei dem Versuch, Bewegung theoretisch zu verstehen, liegt es nahe, von dem Ziel einer Bewegungshandlung auszugehen. Das Bewegungsziel ist abhängig von den Intentionen des Bewegers, die zu diesem Ziel führende Bewegung von den Umweltbedingungen. Allerdings ist es keineswegs so, daß damit die Bewegung schon hinreichend festgelegt wäre, im Gegenteil: Zumeist gibt es unendlich viele verschiedene Bewegungsmöglichkeiten, die unter den gegebenen Umweltbedingungen zu dem gewünschten Bewegungsziel führen. Dabei ist es offensichtlich, daß der Mensch nicht zufällig irgendein Bewegungsmuster auswählt, sondern nach bestimmten Kriterien sein Bewegen „sinnvoll“ organisiert. Das Auswahlproblem wird in der Literatur das „Problem der Freiheitsgrade“ oder (in der Motorikforschung) auch „Bernsteins Problem“ genannt. Unter dem Begriff *action approach* versammeln sich Theorierichtungen, die zum Ziel haben, die Organisationskriterien zum Lösen des Problems der Freiheitsgrade durch die nichtlineare Dynamik des Zusammenwirkens bestimmter Parameter zu erklären.

Zur Verdeutlichung des Problems wird in der Literatur häufig das Beispiel des Ergreifens eines Gegenstandes gewählt. Die Position des Gegenstands als wahrgenommene Umweltbedingung liefert zwar eine Randbedingung, sie legt aber weder die Bahn noch die Geschwindigkeit der Armbewegung fest. Trotzdem kann beobachtet werden, daß stets ähnliche, anscheinend „bequeme“ Bewegungsmuster gewählt werden (z.B. Morasso 1981). Auch im Bereich des Sports gibt es vielfältige Beispiele. So kann beim Tennisaufschlag das Bewegungsziel, den Ball über das Netz in das richtige Feld zu spielen, auf unterschiedlichste Art erreicht werden. Es haben sich aber unter Berücksichtigung der Randbedingung, dem Gegner den Return möglichst zu erschweren, einige verschiedene Techniken (oder Technikvariationen) herausgebildet, die von Könnern immer wieder gewählt werden.

Unter den potentiell unendlich vielen möglichen Bewegungsmustern, die zu einem gewünschten Bewegungsergebnis führen, ein bestimmtes

auszuwählen, heißt, das Problem der Freiheitsgrade zu lösen. Freiheitsgrade der Gelenke werden im sportwissenschaftlichen Lexikon definiert als „die Anzahl der frei wählbaren voneinander unabhängigen Zeitfunktionen, die die räumliche Lage eines Körpers beschreiben“ (Baumann 1992, 169). In dieser Arbeit wird diese Definition in Anlehnung an die vorliegende Literatur nicht nur auf Gelenke, sondern auf alle die Bewegung beeinflussenden Faktoren wie Muskeln, Nerven und deren Bestandteile bezogen.

Das Problem der Freiheitsgrade wurde in dem sich mit der Motorik beschäftigenden wissenschaftlichen Diskurs (eine Sportwissenschaft war noch nicht „erfunden“) zuerst von Nikolai Alexandrejewitsch Bernstein eingebracht. Bernstein war von 1923 bis zu seinem Tod im Jahr 1966 intensiv mit dem Problem der Bewegungsregulation beschäftigt<sup>1</sup>. Für ihn bedeutet Bewegungskoordination nichts anderes als „die Überwindung der überflüssigen Freiheitsgrade des sich bewegenden Organs“ (Bernstein 1988, 181f). An einem Beispiel macht er deutlich, wie schwer die Beherrschung der Freiheitsgrade sein kann:

Befestigen Sie vorn an der Schnalle Ihres Gürtels das obere Ende eines Schistockes. An das Ende, das den Teller trägt, hängen Sie eine Last von 1 bis 2 kg, und an dem Schneeteller befestigen Sie rechts und links je einen Gummischlauch von ausreichender Länge, so daß Sie die Enden jedes Schlauches in die rechte und linke Hand nehmen können. Nachdem Sie den Stock mit der Spitze nach vorn gebracht haben, stellen Sie sich vor eine vertikal stehende Tafel, auf die groß ein Kreis (...) gemalt ist. Nun müssen Sie sich bemühen, durch bloßes Manipulieren mit den Gummischläuchen mit der Stockspitze die aufgezeichnete Figur nachzuziehen. Der Stock stellt hierbei ein Glied einer Extremität mit zwei kinematischen Freiheitsgraden dar, die Schläuche sind zwei antagonistischen Muskeln analog, die in das System zwei weitere, elastische Freiheitsgrade einbringen. Dieser Versuch (...) überzeugt jeden (...), was für eine schwierige und für die Koordination kaum zu bewältigende Sache insgesamt nur vier Freiheitsgrade sind, wenn dem Menschen (...) die motorische Erfahrung nicht zu Diensten steht. (Bernstein 1988, 181).

<sup>1</sup> Außerhalb der UdSSR wurde er jedoch erst durch die Übersetzung seiner wichtigsten Werke ins Englische 1967 bekannt (Bernstein 1967). 1975 erfolgte die Herausgabe eines Sammelbands mit deutschen Übersetzungen, aus dessen zweiter Ausgabe (1988) in dieser Arbeit zitiert wird. Man lasse sich durch die Jahreszahl nicht darüber hinwegtäuschen, daß die zitierten Passagen älteren Ursprungs sind.

Während unter dem Blickwinkel der motorischen Kontrolle die vielen Freiheitsgrade ein Problem bedeuten, ermöglichen sie aus Sicht des sich bewegenden Subjekts die nötige Flexibilität der Handlung. Deshalb spricht Bernstein in der oben zitierten Definition der Koordination auch nicht von Kontrolle, sondern von Überwindung der überflüssigen Freiheitsgrade.

Wie schon oben erwähnt, wird im *action approach* das Problem der Freiheitsgrade durch die Dynamik des Zusammenwirkens einzelner Komponenten gelöst. Dabei unterscheiden sich einzelne Theorien durch die Auswahl der Komponenten und durch den Ordnungsaspekt, unter dem sich das Zusammenwirken organisiert. Durch das Bewegungsziel festgelegte dynamische Kopplungen – von Bernstein *Synergien* genannt – determinieren nach seiner Ansicht die Bewegung. Aus *psychoökologischer* Sicht entsteht Bewegung aus dem Zusammenwirken von Akteur und Umwelt, in der *Synergetik* aus der Dynamik abstrakter Parameter.

### 1.1.1 Bernsteins Synergien

Bernstein führt in seiner Theorie der Koordination den Begriff der *Synergien* ein. Im zweiten Lebenshalbjahr fängt ein Kind an, sinnvolle Bewegungen zielgerichtet durchzuführen, nach Bernstein der Übergang von Synkinesen zu Synergien. „Synkinesen sind gleichzeitige Bewegungen, die keinen sinnvollen Zusammenhang haben und bei Erwachsenen immer pathologisch sind. Die Synergien sind abgestimmte Bewegungen oder ihre Komponenten, die zur gemeinsamen Lösung einer bestimmten Bewegungsaufgabe dienen.“ (Bernstein 1988, 116). Dabei sind Synergien keine festen Kopplungen zwischen den an der Bewegung beteiligten Komponenten, sondern sind an dem Bewegungsziel orientiert. Sie werden im Verlauf des Bewegungslernens dynamisch gebildet. Dabei heben sich nach Bernstein drei Stadien voneinander ab, die sich im bezug auf die Koordination unterscheiden. Im ersten Stadium werden überflüssige Freiheitsgrade dadurch eliminiert, daß nur die für die Bewegungsausführung unbedingt notwendigen Gelenke bewegt werden, die

gesamte restliche Muskulatur wird angespannt, so daß die übrigen Gelenke fixiert werden. Dadurch wirkt die Bewegung „eckig und ungeschickt“. „Die Körperhaltung und die Mimik sind gespannt, die Atmung wird angehalten, die Zunge herausgestreckt usw.“ (a.a.O., 135). Im zweiten Stadium wird die Fixation der Gelenke allmählich zugunsten von kurzen, phasischen Impulsen in die richtige Richtung und zum richtigen Zeitpunkt aufgegeben. Dieses erfordert jedoch die volle Aufmerksamkeit des Lernenden, und häufig ist er noch damit überlastet. Die Kraftkurven zu den Bewegungen sind noch „vielzackig und wenig effektiv“ (ebd.). Das dritte Stadium ist die Könnensstufe. Sie zeichnet sich dadurch aus, daß die Bewegung irritierende Störungen durch reaktive Kräfte aus der Umgebung „in beträchtlichem Umfang zu nützlichen Kräften gemacht werden“, die die Bewegung „in die richtige Bewegungsbahn zurückzuführen suchen“ (ebd.). In der Alltagsmotorik (Gehen, Greifen etc.) werden diese „dynamisch stabilen Bewegungen“ von allen Menschen erlernt, bei sportlichen Bewegungen gilt dies nur für die „besten Meister“ (a.a.O.). Diese von Bernstein intuitiv geschlossenen Lernverläufe wurden erst kürzlich von Vereijken u. a. (1992) empirisch überprüft und bestätigt.

### 1.1.2 Psychoökologie

Der im allgemeinen psychoökologisch (z.B. Wiemeyer 1992; Hossner, Munzert & Olivier 1995), seltener ökopsychologisch (Beek & Sternad 1992) genannte Ansatz beruht auf der von Gibson (1982) formulierten These, daß Bewegung und Wahrnehmung in einer untrennbaren Wechselbeziehung stehen. Bewegungen werden nicht zentral gesteuert, die Existenz zentraler Repräsentationen für Bewegungsmuster wird bezweifelt. Statt dessen wird eine *Akteur-Umwelt-Synergie* (Beek & Sternad 1992, 4) postuliert, aus der sich die Bewegungen ergeben. Einige wenige Variable, die sich aus der Beziehung zwischen Akteur und Umwelt ergeben, bestimmen die zeitliche Koordination der Bewegungen. So werden alle Freiheitsgrade, die zur Festlegung der zeitlichen Koordination determiniert werden müssen, auf diese Variablen reduziert. Eine solche ist beispielsweise die *time to contact*-Variable  $\tau$ , die die Zeit

darstellt, die ein sich dem Auge nähernder Gegenstand bis zum Auftreffen auf die Netzhaut braucht (Lee 1976). Sie wird direkt durch die scheinbare zeitliche Veränderung der Größe des Körpers (bzw. der realen Vergrößerung seines Abbildes auf der Netzhaut) wahrgenommen.

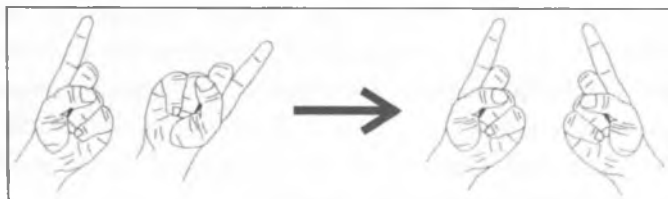
Zur Reduktion der räumlichen Freiheitsgrade übernehmen viele Vertreter des psychoökologischen Ansatzes weitgehend die Idee der Synergien Bernsteins (s. o.) und benennen diese *koordinative Strukturen* (Turvey 1977; Kugler, Kelso & Turvey 1980). Sie werden definiert als „a group of muscles often spanning a number of joints that is constrained to act as a single functional unit“ (Kugler, Kelso & Turvey 1980, 17). Anstelle einer Vielzahl von Variablen, die kontrolliert oder koordiniert werden müssen, wird die Existenz einiger weniger „kollektiver (...) Variablen, die die Invarianzen eines Handlungsvollzugs erfassen“ (Beek & Sternad 1992, 21), angenommen. Die koordinativen Strukturen werden in Abhängigkeit von den Umweltbedingungen „*getunt*“ (a.a.O., 7). Invariante Merkmale gehören zur zugrundeliegenden koordinativen Struktur, während die zum Tuning gehörenden Merkmale einer Änderung unterworfen sind und als *variant* bezeichnet werden. Auch bei der räumlichen Koordination wird also das Problem der Freiheitsgrade durch ein Zusammenwirken von Subjekt und Umwelt gelöst.

### 1.1.3 Synergetik

Synergetik, die Lehre vom Zusammenwirken, ist eine Wissenschaftsdisziplin, die vor ungefähr 20 Jahren von Hermann Haken ins Leben gerufen wurde. Ihre zentrale Frage ist die nach der Existenz „einheitlicher Prinzipien für das Entstehen von Strukturen, unabhängig von der Art der einzelnen Teile, aus denen diese bestehen“ (Haken 1992, 54). Haken beschreibt dabei diese Strukturen mit einem (oder mehreren) *Ordnungsparameter*, der die einzelnen Teile *versklavt*. Der Ordnungsparameter ist eine dynamische Größe, dessen dynamisches Verhalten mathematisch exakt definierbar ist. Anschaulich entspricht es dem Verhalten einer Kugel auf einer (durch die Gleichung definierten) Potentiallandschaft. Die Täler dieser Landschaft (lokale Minima der

Funktion) werden als *Attraktoren* bezeichnet. So wie die Kugel in das Tal rollt, werden die einzelnen Teile von dem durch den Attraktor spezifizierten Zustand angezogen. Das Aussehen der Potentiallandschaft und damit die Dynamik des Ordnungsparameters wird durch einen oder mehrere *Kontrollparameter* bestimmt.

Solche Ordnungs- und Kontrollparameter wurden nun auch für die Beschreibung von Bewegungsverhalten gesucht. Dies gelang zuerst für zyklische Fingerbewegungen (Haken, Kelso & Bunz 1985; Schöner, Haken & Kelso 1986). In einem Experiment sollten Versuchspersonen die Zeigefinger ihrer beiden Hände in einer vorgegebenen Frequenz parallel vor sich bewegen. Wurde die Frequenz erhöht, fand zu einem bestimmten Zeitpunkt ein spontaner Wechsel zur achsensymmetrischen Bewegung statt (s. Abb. 1).



**Abb. 1:** Der Übergang von der parallelen (links) zur antiparallelen, d.h. symmetrischen Bewegung (rechts) (nach Haken 1992, 61). Im Originalexperiment wurden Fingerbewegungen des Zeigefingers untersucht.

Ordnungsparameter ist hier die relative Frequenz zwischen den beiden Fingern, Kontrollparameter die Bewegungsfrequenz. Das empirisch gefundene Bewegungsverhalten kann so recht genau modelliert werden, sogar Details im Bewegungsverlauf während des Übergangs vom parallelen zum symmetrischen Zustand können durch hier nicht erläuterte theoretische Gesetzmäßigkeiten der Synergetik (*critical slowing-down*, *critical fluctuations*) vorhergesagt werden. Bezogen auf das Problem der Freiheitsgrade fassen Schöner, Haken und Kelso (1986, 256) zusammen:

Were one to attempt to count the neurons, neuronal connections, vascular support processes, muscles, and joints that cooperate to produce these "simple" rhythmical motions of the limbs, the result would be a very large number of degrees of freedom. The present paper, however, (...) shows that this high dimensionality can be compressed into a single order parameter that obeys a simple order parameter equation.

Neuerdings wird die Existenz von Ordnungsparametern untermauert, indem gezeigt wird, daß sie auch zur dynamischen Mustererkennung verwendet werden (Haas u.a. 1995).

## 1.2 Modellierung auf der algorithmischen Ebene: Der *motor approach*

Mit der kognitiven Wende Ende der 50er, Anfang der 60er Jahre unseres Jahrhunderts werden der Behaviorismus und die Gestaltpsychologie, die bis dahin vorherrschenden Schulen in der Psychologie, verdrängt durch eine neue Sichtweise, die „Psychologie der Informationsverarbeitung“, auch „Kognitive Psychologie“ oder „Kognitivismus“ genannt. Dabei wird angenommen, daß mentale Prozesse des Menschen auf denselben Prinzipien beruhen wie informations- bzw. symbolverarbeitende Prozesse eines Computers, man spricht von der „Computer-Metapher“ oder der „Computer-Analogie“. Bei der Kognitiven Psychologie handelt es sich nicht um eine psychologische Schule, deren Anhänger durch eine gemeinsame *Theorie* geeint werden, sondern vielmehr um eine „Strömung“, die sich auf die *Computer-Metapher* stützt (vgl. Neumann & Prinz 1990, 14). Diese Strömung wird zum *mainstream* der psychologischen Forschung. Zentrale Begriffe des Behaviorismus wie „Reiz“, „Reaktion“ und „Verstärkung“ werden abgelöst durch „Input“, „Output“ und „Speichern“ (vgl. Holzkamp 1993, 118f). Durch die Verwendung der Computer-Metapher können die zuvor von naturwissenschaftlichen Psychologen als „schwammig“ verpönten mentalen Prozesse exakt beschrieben werden.

Psychische Prozesse als Informationsverarbeitungsprozesse zu betrachten und zu verstehen, bietet nach der Auffassung vieler Psychologen zum ersten

Mal die Chance, tatsächlich eine klare Sprache für die Beschreibung psychischer Zustände und Prozesse zu finden (Dörner 1991, 39).

Die Sprache wird aus der Informatik, und hier speziell aus dem Bereich der Künstlichen Intelligenz (KI), übernommen.

Diese Entwicklung bleibt nicht ohne Konsequenzen für die Erforschung sportlicher Bewegungen. Die Forschung in der Bewegungswissenschaft ist angewiesen auf Modelle und Theorien aus der Grundlagenforschung. Im Labor gefundene (psychologische) Theorien über Bewegungskontrolle und Bewegungslernen werden im sportwissenschaftlichen Kontext auf ihre Gültigkeit für komplexe, großmotorische Bewegungen aus der Sportpraxis überprüft.

Unter den Theorieansätzen zur Motorikforschung, die sich auf die Computer-Metapher beziehen (z.B. Keele 1968; Adams 1971), hat sich die Theorie der durch ein Schema parametrisierten generalisierten motorischen Programme (GMP-/Schematheorie) von Schmidt (1975/1990) „als Synonym für Repräsentationsansätze in der Motorikforschung schlechthin“ (Munzert 1992, 346) durchgesetzt. Sie wurde schon vielfach in sportwissenschaftlichen Werken dargestellt (z.B. Willimczik & Roth 1983; Roth 1989; Wiemeyer 1992) und soll deswegen hier nur kurz umrissen werden.

In seiner Theorie postuliert Schmidt die Existenz generalisierter motorischer Programme (GMPs) zur Steuerung von Bewegungen. Für jede Bewegungsklasse gibt es ein GMP. Alle Bewegungen innerhalb einer Klasse haben die gleiche Gestalt (Gestaltkonstanz-Hypothese). Sie wird bestimmt durch interne *Programminvarianten*, welche die relativen Abstände, die relativen Zeiten und die relativen Höhen der Muskelaktivierung (*sequencing, relative timing, relative forces*) festlegen. Für die Ausführung einer konkreten Bewegung aus einer Bewegungsklasse wird nach der Auswahl des GMPs die zeitlich-dynamische Ausprägung der Bewegungsgestalt durch *Parameter* spezifiziert. Die wichtigsten Parameter sind die Bewegungsdauer und der Gesamtkrafteinsatz. Ihre Auswahl wird durch das *Wiedergabeschema (recall schema)* geleistet und hängt von den Anfangsbedingungen und dem gewünschten Ergebnis der Bewegung ab. Nach erfolgter Bewegung wird die Beziehung



zwischen der Ausgangsbedingung, dem tatsächlichen Ergebnis und den dazugehörigen Parametern gespeichert. Das Wiedergabeschema entsteht als die Abstraktion verschiedener solcher Beziehungen zu einer Regelgeraden<sup>2</sup>.

Neben dem Wiedergabeschema, das für die Bewegungsproduktion verantwortlich ist, gibt es das Wiedererkennungsschema (*recognition schema*) zur Bewegungsevaluation. Es wird durch die Abstraktion der verschiedenen Beziehungen zwischen Ausgangsbedingungen, tatsächlichen Ergebnissen und sensorischen Konsequenzen gebildet.

### 1.3 Modellierung auf der implementationellen Ebene

Bei der Entwicklung von Robotern stellt sich in den Ingenieurwissenschaften das Problem der konkreten Implementation von Steuerungsbefehlen an Bewegung erzeugende Motoren. Hier treffen die Entwickler wiederum auf das Problem der Freiheitsgrade. Auch für einen Roboter existieren viele verschiedene Möglichkeiten, eine Bewegungsbahn von einem Punkt zu einem anderen zu vollführen, wie zum Beispiel beim Greifen eines Gegenstands. In der ingenieurwissenschaftlichen Sprachweise unterscheidet man hier zwischen korrekt gestellten (*well-posed*) und unkorrekt gestellten (*ill-posed*) Problemen.

Ein Problem ist korrekt gestellt, wenn eine einzige Lösung existiert, die kontinuierlich von den Eingangsdaten abhängt, wenn also alle Unbekannten durch ein Gleichungssystem aus Bewegungsgleichungen und Randbedingungen bestimmt werden können und so kein überflüssiger Freiheitsgrad vorhanden ist. Andernfalls ist es unkorrekt gestellt und hat so viele Freiheitsgrade wie unbestimmbare unabhängige Variablen. Die Probleme motorischer Kontrolle haben viele Lösungen und sind daher unkorrekt gestellt. Dabei tritt das Problem der Freiheitsgrade in drei

<sup>2</sup> Richtiger ist es, von einer Regelebene zu sprechen, da Ausgangsbedingung und Ergebnis der Bewegung verschiedene Dimensionen darstellen (vgl. Hossner 1991, 13). Stellt man die absolute Dauer und den Gesamtkrafteinsatz (die beiden wichtigsten Parameter) auch noch in getrennten Dimensionen dar, ist man schon beim Regelwürfel.

Ebenen auf:

- Die Form der Bewegungsbahn wird nicht durch den Start- und den Zielpunkt festgelegt. Hier lassen sich unendlich viele Bahnen vorstellen (Abbildung 2a).
- Das Problem der inversen Kinematik ist das Problem, für einen gegebenen Punkt im Raum die verschiedenen Gelenkwinkel zu bestimmen.

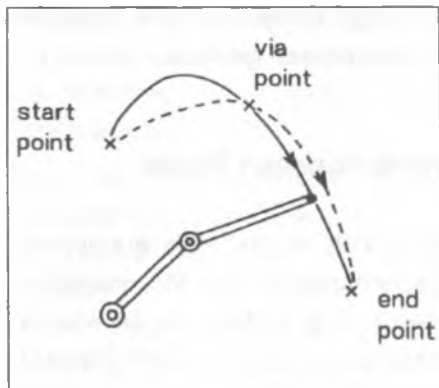


Abb. 2a: Freiheitsgrade bei der Wahl der Bewegungsbahn (aus Kawato 1990a, 200).

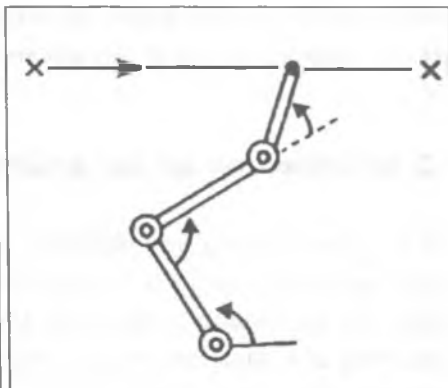


Abb. 2b: Freiheitsgrade bei der Wahl der Gelenkwinkel für eine vorgegebene Position (aus Kawato 1990a, 200).

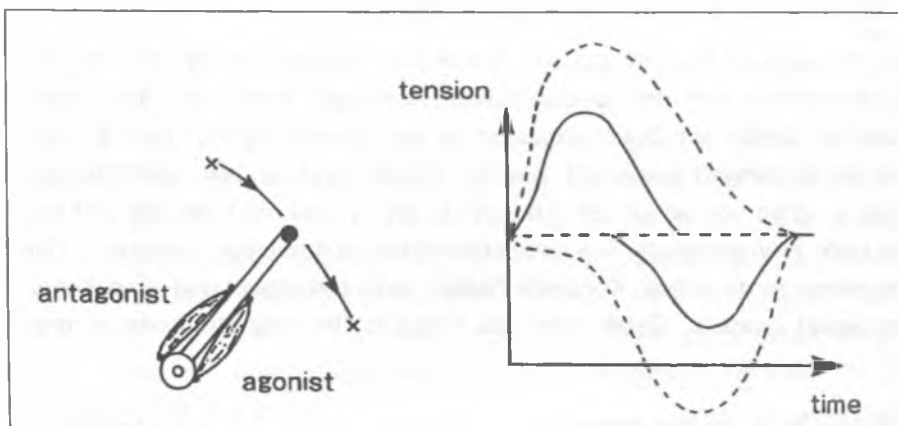


Abb. 2c: Freiheitsgrade bei der Wahl der Muskelaktivität für ein vorgegebenes Drehmoment (aus Kawato 1990a, 200).

Es ist die Umkehrung der Aufgabe, aus der Stellung der Gelenkwinkel die Position des Endglieds zu errechnen. Während letzteres Problem immer korrekt gestellt ist, gibt es, wenn die Anzahl der Freiheitsgrade der Gelenke die Anzahl der durch die Randbedingungen festgelegten Freiheitsgrade übersteigt, für das inverse kinematische Problem mehrere Lösungen (Abbildung 2b).

- Das Problem der inversen Dynamik ist das Problem, für eine Drehbewegung im Gelenk die richtigen Kräfte zu finden. Auch hier lassen sich viele verschiedene Kräftepaare von Agonist und Antagonist für das gleiche Drehmoment finden. Wenn noch mehr als zwei Muskeln die Bewegung bestimmen, wird dieses Problem noch komplizierter (Abbildung 2c).

Auf der Implementationsebene geht es bei der Bewegungssteuerung um die Frage nach dem konkreten Inhalt der Steuerungssignale. Die Lösung des Problems der Freiheitsgrade mit Hilfe von Optimierungskriterien postuliert, daß auf der Implementationsebene die Steuerungssignale bestimmten Kriterien genügen müssen; die *mass-spring*-Hypothese geht davon aus, daß sie durch Federkonstanten modellierbar sind. Dabei spielt es im Prinzip keine Rolle, ob die Optimierung bzw. die Federkonstanten nun durch Programme errechnet werden oder durch dynamisches Zusammenspiel einzelner Komponenten entstehen.

### 1.3.1 Optimierungsverfahren

Ein aus der Robotik stammendes Verfahren, die überflüssigen Freiheitsgrade zu überwinden, besteht in der Formulierung von Kriterien, die eine optimale Bewegung charakterisieren. Solche Optimierungskriterien sind Randbedingungen auf der Implementationsebene. Sie werden durch Beobachten der Muskelaktivierung bzw. der Beschleunigung im Bewegungsverlauf während des aktuellen Bewegungsvollzugs gefunden. Alle möglichen Bewegungsmuster können nach diesem Ansatz auf ein Kriterium hin überprüft werden, und dasjenige, welches dieses Kriterium optimiert, wird dann ausgeführt.

Flash und Hogan (1985) schlagen in ihrem Modell des *minimalen Rucks*

(*minimum jerk model*) vor, daß unter allen Bewegungsmöglichkeiten diejenige mit der geringsten Änderung im Beschleunigungsverlauf (Ruck), integriert über den gesamten Bewegungsverlauf, natürliches Bewegungsverhalten am ehesten modelliert. Für einen zweidimensionalen Raum führt das zu der Formel

$$C_J = \int_0^{t_{end}} \left( \left( \frac{d^3 x}{dt} \right)^2 + \left( \frac{d^3 y}{dt} \right)^2 \right) dt \quad (1)$$

Ein verwandtes Modell, das jedoch natürliche (Arm-)Bewegungen vor allem seitlich des Körpers besser beschreibt und außerdem auf Drehmomenten (und damit Kräften) und nicht auf Beschleunigungen beruht, ist das Modell der *minimalen Änderung der Drehmomente* (*minimum torque change model*) von Uno, Kawato & Suzuki (1989). Über die gesamte Bewegung soll die Summe des Quadrats der Änderung aller Drehmomente  $T$  der einzelnen Muskeln  $i$  aus  $n$  minimiert werden. In einer Formel:

$$C_T = \int_0^{t_{end}} \sum_{i=1}^n \left( \frac{dT_i}{dt} \right)^2 dt \quad (2)$$

Die Lösung des Problems der Freiheitsgrade wird hier also durch ein explizit berechnetes Optimierungsverfahren gefunden.

### 1.3.2 Die *mass-spring*-Hypothese

Eine elegante Lösung des Problems der Freiheitsgrade für Positionierungsbewegungen bietet die Modellierung des sich bewegenden organischen Systems durch ein System aus *Masse* und (komplizierten nichtlinearen) *Federn* (Asatryan & Feldman 1965, Feldman 1966a, 1966b). Hierbei werden im einfachsten Fall eines eingelenkigen Systems Agonist und Antagonist durch jeweils eine Feder ersetzt, der Endeffektor durch eine Masse, der Knochen durch eine starre Stange, und das Gelenk bleibt ein Gelenk (s. Abb. 3).

Soll nun eine bestimmte Position im Raum erreicht werden, werden die Federkonstanten so eingestellt, daß sich die Masse in der zu erreichen-

den Position im *Gleichgewicht* befindet. Dieses wird sich dann auf natürliche Weise dynamisch einstellen. Dabei muß weder die Ausgangsposition spezifiziert noch die Bewegungsbahn vorgegeben werden. Auch externe Störungen im Bewegungsablauf müssen nicht durch neue Einstellungen korrigiert werden. Somit wird die Anzahl der Freiheitsgrade beschränkt auf die Anzahl der „Federn“, deren Federkonstante für einen Gleichgewichtszustand eingestellt werden müssen.

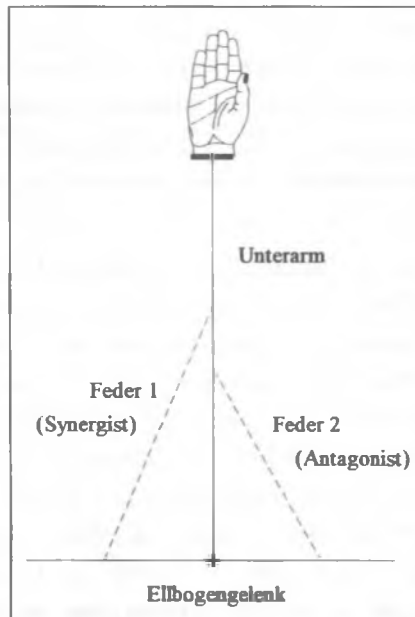


Abb. 3: Der menschliche Unterarm im *mass-spring*-Modell.

Die Zugehörigkeit zur Implementationsebene wird daran deutlich, daß der Masse-Feder-Mechanismus allen Theorierichtungen zugeordnet werden kann: Während beispielsweise Roth (1991, 7) die zu den jeweiligen Positionen gehörenden Federkonstanten als Inhalte eines motorischen Programms bezeichnet, behandelt Schmidt (1988a, 5ff.) die *mass-spring*-Hypothese unter der Überschrift: „The action perspective –

a brief description“ und stellt dabei die funktionellen dynamischen Eigenschaften der Muskeln in den Vordergrund.

#### 1.4 Stärken und Schwächen der verschiedenen Herangehensweisen

Der *motor approach* als die in den letzten 20 Jahren dominierende Herangehensweise an das Problem der Erklärung menschlicher Willkürbewegungen konnte seine Stärken am besten entwickeln, stand aber ebenso im Zentrum der Kritik. Die alternativen Ansätze definieren sich in Abgrenzung zu diesem *mainstream* der Motorikforschung. Ihre Stärken erwachsen aus den Schwächen des *motor approach*, wobei dessen Stärken häufig ihre Schwächen sind. So sollen also Vorzüge und Nachteile dieses Ansatzes im Zentrum dieses Kapitels stehen.

Wichtigste Stärke dieser prozeßorientierten Herangehensweise ist, daß deren wohldefinierte Konstrukte die Ableitung von Hypothesen erlaubt, die in Experimenten empirisch überprüft werden können. So kann beispielsweise auf der Grundlage von Schmidts Theorie die Hypothese abgeleitet werden, daß durch ein Variieren bestimmter Übungsbedingungen regelmäßige Zusammenhänge zwischen der Bewegung, ihren Ausgangsbedingungen und ihrem Ergebnis besser gelernt wird als wenn die Übungsbedingungen konstant gehalten werden. Dieses Lernen der Zusammenhänge erlaubt dann einen Transfer zu bisher noch nicht ausgeführten Bewegungen. Im Zusammenhang mit der in dieser Arbeit durchgeführten Simulation wird diese Hypothese und ein alternativer Erklärungsansatz ausführlicher dargestellt (vgl. Kapitel 4.2, S. 83 ff.).

Die genaue Definierbarkeit der Konstrukte, die empirische Untersuchungen erst ermöglicht, hat aber auch ihren Preis. Alle Phänomene oder Prozesse, die über den Definitionsbereich hinausgehen, können nicht erklärt werden. So bleibt bei dem *motor approach* unbefriedigend, daß keine Aussage über Auswahl und Erwerb der Programme gemacht wird, was Schmidt (1988b, 483) auch offen zugibt:

However, like most programming theories, the schema theory does not specify from where the motor programs come. This is an important problem, but the level of knowledge at this time does not allow much to be said about this process.

Das Problem ist eine logische Konsequenz aus der Computer-Analogie. Dem Computer werden die Programme von seinem Benutzer eingegeben, bei der menschlichen Informationsverarbeitung fehlt ein Analogon zu diesem Benutzer. Die Fähigkeit des Menschen, sich bewußt zu entscheiden, *was* er lernen möchte und *wie* er handelt, liegt außerhalb des Erkenntnisinteresses der Programmtheoretiker, die Ausklammerung führt aber zu dem Vorwurf,

that the embodiment of the details of the actions in a programme or other central representation simply pushes the problem of coordinations deeper into the system, resulting in a logically infinite regress (Schmidt 1988a, 6, einen Kritikpunkt an dem *motor approach* von Kugler, Kelso & Turvey 1980 zusammenfassend).

Eine zweite Richtung der Kritik argumentiert mit dem Problem der Freiheitsgrade. Unter dem Blickwinkel des Informationsverarbeitungsansatzes wird dem Problem durch Speichern des GMP und des dazugehörigen Schemas begegnet. Mit der Auswahl des Programms und der entsprechenden Parameter ist die eine Bewegung prinzipiell festgelegt, es gibt dann keine Freiheitsgrade mehr. Dabei entsteht das *Freiheitsgrade-Speicherplatz-Dilemma*. Je höher die Freiheitsgrade des motorischen Systems, um so mehr Programme werden für die Steuerung gebraucht und um so größer muß folglich der Speicherplatz sein.

Es ist offensichtlich, daß das Freiheitsgrade-Speicherplatz-Dilemma mit Schmidts Theorie nicht grundsätzlich gelöst wird. Die Bewegungsklassen, innerhalb derer sich die relativen Kräfte und das relative Timing linear verhalten, sind denkbar klein und keinesfalls mit den Bewegungsklassen der sportlichen Terminologie vergleichbar. Je enger jedoch die Bewegungsklasse ist, desto größer ist die Anzahl der generalisierten motorischen Programme und der dazugehörigen Schemata. Es muß sogar bezweifelt werden, ob es überhaupt verschiedene Bewegungen gibt, deren Kraft- und Zeitparameter linear voneinander abhängen und so eine Bewegungsklasse bilden. Aus der Robotik ist bekannt, daß

Probleme der inversen Kinematik und inversen Dynamik nur mit Näherungen von nichtlinearen Differentialgleichungen und Optimierungskriterien als Randbedingungen zu lösen sind. Auch für die menschliche Bewegung wurde nachgewiesen, daß bei einer höheren Ausführungsgeschwindigkeit die Kraft- und Zeitverlaufskurven sich nicht linear ändern (für Armbewegungen: Schneider, Zernicke, Schmidt & Hart 1989; für Strampelbewegungen von Säuglingen: Schneider, Zernicke, Ulrich, Jensen & Thelen 1990), ja daß sich sogar beim Üben von einfachen schnellen Armbewegungen in der gleichen Ausführungsgeschwindigkeit Kraft- und Zeitverlaufskurven während des Übens verändern (Schneider u.a. 1989; Schneider 1992), was nach Schmidts (1975/1990, 1988b) Definition einer Programmänderung gleichkäme.

Wenn man die von Schmidt formulierten linearen Zusammenhänge ernst nimmt, wächst die Zahl der GMPs und der dazugehörigen Schemata rasch ins Unermeßliche. Ließe man jedoch innerhalb einer Bewegungsklasse *nichtlineare* Parameteränderungen zu, erhöhte dies die Breite der Bewegungsklassen, und das Speicherproblem wäre gelöst.

Genau auf dieser Idee der nichtlinearen Zusammenhänge beruhen die alternativen Herangehensweisen an die Bewegungskontrolle und das Problem der Freiheitsgrade. Leider entziehen sie sich dadurch zumeist der formalen Analyse, denn nichtlineare Differentialgleichungen sind mathematisch schwierig zu behandeln, oft sogar „unlösbar“. Nachteilig an nichtlinearen Modellen ist daher, daß sie empirisch meist nicht überprüft werden können. Häufig sind die in das Modell eingehenden Parameter qualitativ und quantitativ nicht spezifizierbar, und es kann nicht vorhergesehen werden, wie das System bei Änderung einzelner Parameter reagiert, da „alles mit allem zusammenhängt“ und „das Ganze mehr ist als die Summe der Teile“.

Durch ihre Konzentration auf die komputationelle Ebene sind für den *action approach* eine Reihe von empirischen Befunden problematisch, die die Relevanz von zentralen Repräsentationen, also die Bedeutung der algorithmischen Ebene, für die Bewegungsproduktion nahelegen.



- In Deafferentiationsstudien wurde nachgewiesen, daß auch ohne afferente Signale Bewegungen relativ präzise ausgeführt werden können (z. B. Lashley 1917; Kelso 1977).
- Bewegungen lassen sich erlernen. Zu verschiedenen Zeitpunkten kann eine Bewegung einmal stabil schlechter und das andere Mal stabil besser ausgeführt werden.
- Lerneffekte lassen sich sogar bei mentalem Training nachweisen (Heuer 1985). Eine Verbesserung der Bewegungsqualität erfolgt alleine dadurch, daß der Bewegungsablauf „im Geiste“ nachvollzogen wurde.

Die Hauptaussage des *action approach*, daß Bewegungen nicht zentral gesteuert werden, wird durch diese Befunde scharf angegriffen.

Für den Masse-Feder-Ansatz ist das Modellieren verschiedener Bewegungsgeschwindigkeiten sowie das „Einfrieren“ der Bewegung recht kompliziert, und die menschliche Fähigkeit zur willkürlichen Spezifikation von Bewegungsbahnen kann nicht erklärt werden. Lösungsversuche, die mehrere oder alle Punkte auf der Bewegungsbahn als anzusteuernde Gleichgewichtspunkte vorsehen (z. B. Feldman 1986), führen das Problem der Freiheitsgrade auf der Ebene der Wahl der Bewegungsbahn wieder ein.

Optimierungsverfahren schließlich sind erst dann sinnvoll zu gebrauchen, wenn ein effizienter Weg gefunden wird, die formulierten Kriterien zu erfüllen.

## 1.5 Warum eine neue Theorie?

Ohne die Verdienste aller Ansätze um die theoretische Behandlung menschlicher Bewegungen schmälern zu wollen, soll hier begründet werden, warum eine neue theoretische Herangehensweise not tut. Die sportbezogene Bewegungswissenschaft will beschreiben und erklären, wie motorisches Lernen und motorische Kontrolle bei sportlichen Bewegungen ablaufen. Dafür werden zumeist (wie schon oben erwähnt)

empirische Befunde psychomotorischer Experimente auf ihre Validität in der Sportpraxis überprüft.

Die Erkenntnisse, die der informationstheoretische Ansatz für die Sportpraxis geliefert hat, erscheinen doch merkwürdig verkürzt. Aus vier Gründen können prinzipiell von einer kognitivistisch geprägten Bewegungswissenschaft nur beschränkt gültige Aussagen für die Sportpraxis erwartet werden:

- Mag die Computer-Metapher schon für kognitive Prozesse zweifelhaft sein, so ist sie für die Peripherie gänzlich untauglich. Menschliche Bewegungen haben nun wirklich nichts, was eine Analogie zwischen dem sich bewegenden Körper und Drucker oder Bildschirm rechtfertigen könnte (vgl. Neumann 1985, 30; 1993, 62f).
- Vor allem neuere, aus der KI stammende psychologische Modelle auf der Grundlage des Informationsverarbeitungsparadigmas fassen mentale Prozesse als sequentielles Abarbeiten logischer Operationen auf. Es scheint jedoch sehr zweifelhaft, ob dies eine adäquate Herangehensweise an Probleme der motorischen Kontrolle ist.
- Lernen hat im Kognitivismus kaum eine selbständige konzeptionelle Bedeutung. An seine Stelle tritt das Gedächtnis als „kognitivistische Fassung oder Spezifizierung des SR-theoretischen [d.h. behavioristischen] Lernkonzeptes“ (Holzkamp 1993, 120). So liegt der Schwerpunkt des Interesses der Kognitiven Psychologie auf dem Speichern und Abrufen von Informationen, wenig interessiert jedoch der Erwerb von (motorischen) Fertigkeiten. Gerade dies ist jedoch eines der wichtigsten Interessen sportwissenschaftlicher Bewegungsforschung.
- Sowohl der Behaviorismus als auch der Kognitivismus folgten der Maxime „*don't physiologize*“ (Neumann 1993, 61). Es erscheint aber durchaus plausibel, daß physiologische Eigenschaften bei der Beschreibung und Erklärung von Bewegungsverhalten von einiger Bedeutung sind. So fordert Hossner (im Druck) „*please physiologize!*“ für die weitere bewegungswissenschaftliche Forschung.

Die Aussagekraft des *action approach* zu Problemen der Sportpraxis ist

noch geringer. Da Lernen stark mit der Annahme zentraler Repräsentationen zusammenhängt, wird es von Forschern dieser Richtung nicht thematisiert. Eine Schwierigkeit für die sportwissenschaftliche Verortung dieses Ansatzes besteht auch darin, daß es darin kein Äquivalent zu der algorithmischen Ebene des Informationsverarbeitungsansatzes gibt. Gerade die in dieser Ebene aufgeworfenen Fragen nach Reaktions- und Verarbeitungszeiten sowie störende oder beeinflussende Faktoren beim menschlichen Bewegungsverhalten sind auch im sportwissenschaftlichen Zusammenhang spannend.

Zunehmend stellen sich also Fragen, die nur mit einem Modell gelöst werden können, das die Motorik nicht als „peripher“ außen vor läßt, sondern mit dem motorischen Lernen konzeptionell in die Theoriebildung mit einbezogen wird, das physiologische Eigenschaften abbildet und mit dem konkrete Aussagen zur Bewegungsausführung möglich sind. Im folgenden Teil der Arbeit wird der Frage nachgegangen, ob das künstliche neuronale Netz – das Modell des Konnektionismus – diese Anforderungen erfüllen kann.

## 2 Was ist Konnektionismus?

Konnektionismus ist die Bezeichnung für ein neu entstandenes Forschungsparadigma, das Psychologen, Biologen, Neurophysiologen und Informatiker eint. „There are conferences and new books [about connectionism] nearly every day, and the popular science press hails this new wave of theorizing as a breakthrough in understanding the mind“, müssen Fodor und Pylyshyn (1988, 4) als Gegner des Konnektionismus neidvoll eingestehen. In diesem Kapitel der Arbeit sollen die historischen Wurzeln und die grundlegenden Prinzipien des Konnektionismus dargelegt werden.

### 2.1 Definitionen und Terminologie

Der Name „Konnektionismus“ wurde von Thorndike zuerst in die Wissenschaft eingeführt, im Laufe der letzten Jahre hat sich die Bedeutung dieses Begriffs jedoch grundsätzlich gewandelt. Dies hat sich in den gängigen Lexika noch nicht niedergeschlagen, lediglich im McMillan Dictionary of Psychology (Sutherland 1989) wird Konnektionismus wie folgt umschrieben:

Nowadays a synonym for parallel distributed processing, and rapidly becoming preferred to that expression. The term goes back to Thorndike, who used it to describe connections formed between stimuli and responses.

Durch die Vielzahl von Verbindungen (Konnexionen) ergibt sich die Möglichkeit und Notwendigkeit einer hochparallelen und verteilten Verarbeitung. „Parallele verteilte Prozesse“ sind als Konsequenz aus konnektionistischen Modellen zu betrachten und werden auch in dieser Arbeit synonym dazu verwandt.

Dieses Modell der *Parallel Distributed Processes* (PDP) nimmt an, daß

information processing takes place through the interactions of a large number of simple processing elements called units, each sending excitatory and inhibitory signals to other units (McClelland, Rumelhart & Hinton 1986, S. 10).

Verbindendes Merkmal aller konnektionistischen Modelle sind Computersimulationen mit künstlichen neuronalen Netzwerken. Im folgenden sollen die Begriffe „Netz“ und „Netzwerk“ synonym behandelt werden (wie auch in der amerikanischen Literatur mit „*net*“ und „*network*“ üblich), und wenn nicht explizit angegeben, handelt es sich in dieser Arbeit immer um künstliche neuronale Netzwerke. Die „einfachen Verarbeitungselemente“ aus obigem Zitat (auch „Einheiten“ oder „Zellen“), aus denen neuronale Netzwerke bestehen, werden hier konsequenterweise auch „Neuronen“ genannt. Dabei soll jedoch nicht die Tatsache unterschlagen werden, daß es sich um einfachste abstrakte Modelle eines biologischen Neurons handelt, deren Funktionsweise prinzipiell, deren Komplexität aber nicht annähernd abgebildet wird (vgl. z.B. Creutzfeld 1991; Hucho 1991).

## 2.2 Historische Wurzeln

Dem heutigen, modernen Konnektionismus ähnliche Modelle wurden schon in einer „vorpsychologischen“ Zeit bei Spencer in seinen „Principien der Psychologie“ (1903, Original 1855) beschrieben. „For Spencer (...) all subjective mental phenomena, including those of reasoning as well as perception, were to be related to neural mechanisms“ (Walker 1992, 125). Die von Spencer vorgeschlagenen Assoziationsnetze (Theil V, Capitel III-V) haben schon eine erstaunliche Ähnlichkeit zu heute aktuellen neuronalen Netzwerken. James präzierte in seinem Werk „Psychology“ (1890) Spencers Ideen über die „Gedanken“ zu folgenden Annahmen:

When two elementary brain-processes have been active together or in immediate succession, one of them, on re-occurring, tends to propagate its excitement into the other. (...) The amount of activity at any given point in the brain-cortex is the sum of the tendencies of all other points to discharge into it, such tendencies being proportionate (1) to the number of times the excitement of each other point may have accompanied that of the point in question; (2) to the intensity of such excitements; and (3) to the absence of any rival point functionally disconnected with the first point, into which the discharges might be diverted (James 1890/1988, 5).

Aufgrund fehlender Theorien blieb jedoch die Auswirkung des Zusammenspiels der Neuronen nahezu beliebig. So folgerte James daraus auch soziologische Thesen wie die Existenz eines Habitus', der die Aufrechterhaltung der aristokratischen Herrschaftsverhältnisse sicherte. Diese Beliebigkeit und ihre unpopulär werdenden soziologischen Folgerungen diskreditierten James' Ansatz.

Obwohl Thorndike, einer der Gründer des Behaviorismus und Schüler von James, sich selbst einen Konnektionisten nannte (Walker 1992, 150), wird in seiner Theorie über das Konditionieren von Tieren aber deutlich, daß „neural connections are becoming a substitute for, rather than an mechanism of, ideational processes“ (ebd.). Die weitere Entwicklung des Behaviorismus und das Bestreben der Psychologen, naturwissenschaftliche Forschungsprogramme zu übernehmen, führten zur Ausklammerung kognitiver und motivationaler Prozesse. So wurde es kennzeichnend für den Behavioristen, daß er die Psychologie als eine Wissenschaft verstand, die sich vor allem oder gar ausschließlich mit dem äußerlich sichtbaren Verhalten befassen sollte. Das Gehirn des Menschen wurde als eine *black box* betrachtet. Somit wurden auch konnektionistische Ideen im heutigen Sinne, die sich ausdrücklich mit kognitiven Aspekten beschäftigen, ausgeblendet.

Die kognitive Wende führte, wie schon in Kapitel 1 erwähnt, dazu, daß sich die Psychologie wieder mehr den mentalen Prozessen zuwandte. Die Künstliche-Intelligenz-Forschung, an deren Konzept, Sprache und Methode sich die Kognitive Psychologie anlehnte, war anfangs in zwei gleichberechtigte Lager geteilt, eines, das Wissen durch Symbole und formale Regeln implementierte, das zweite, das sich an einfachen Modellen des Gehirns orientierte. McCulloch und Pitts, die zu dem zweiten Lager gehörten, entwarfen 1943 ein mathematisches Modell des zentralen Nervensystems. Es war der erste Versuch, die im Hirn ablaufenden neuronalen Interaktionsprozesse, soweit damals bekannt, mathematisch zu modellieren. Dazu entwickelten sie ein Element, das die Eigenschaften eines Neurons und seiner Verbindungen stark vereinfacht abstrakt modelliert. Dieses Element dient auch den heutigen Netzwerkmodellierern als Grundlage und wird McCulloch-Pitts-Zelle

genannt.

Netze aus McCulloch-Pitts-Zellen waren Gegenstand mathematischen Forschungsinteresses. Pitts und McCulloch (1947) bewiesen, daß mit der geeigneten Wahl von Schwellenwerten und Verbindungen jeder endliche logische Ausdruck berechnet werden kann, was damals großes Aufsehen erregte. Auch Psychologen interessierten sich wieder für neuronale Netze. Hebb entwickelte 1949 eine Idee für das Zusammenwirken einzelner Neuronen:

When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased (Hebb 1949, 50).

Eine vereinfachte Version, formuliert für Lernmechanismen neuronaler Netze, lautet: „When unit A and unit B are simultaneously excited, increase the strength of the connection between them“ (McClelland, Rumelhart & Hinton 1986, 36). Diese einfache Regel zur Bestimmung der Verbindungsstärke zweier Neuronen über die Zeit wurde von Mathematikern und Informatikern weiterentwickelt und gilt als Grundlage für die wichtigsten „Lernregeln“ in neuronalen Netzen.

Lashley formulierte den Grundgedanken der verteilten Wissensrepräsentation. Er scheiterte bei der Suche nach einem Engramm im Hirn und stellte fest, daß „there are no special cells reserved for special memories“ (Lashley 1950, 500).

1958 entwickelte der amerikanische Psychologe Frank Rosenblatt ein neuronales Netz zur Modellierung der visuellen Wahrnehmung (Rosenblatt 1958). Durch einen Lernalgorithmus, der unüberwachtes, selbstorganisiertes Lernen ermöglicht, waren seine *Perceptrons* in der Lage, Kanten in einem präsentierten Bild zu erkennen. Diese Entwicklung stellte eine Herausforderung für Psychologen, Ingenieure und auch Mathematiker dar. Die Forschung über das Perceptron und seine bald entwickelten Varianten blühte, jedoch stellte sich heraus, daß mit größeren Systemen auch die Koordinationsprobleme innerhalb des Netzes größer wurden. Minsky und Papert erstellten 1969 in ihrem Buch *Perceptrons* eine brillante und umfassende Analyse von mathe-

matisch beweisbaren und auch intuitiv geschlossenen Beschränkungen des Perceptrons und leiteten damit die *dark ages* ein, „where, suddenly, research on neural networks was unloved, unwanted, and, most important, unfunded“ (Anderson & Rosenfeld 1988, S. 157).

Erst um 1980 setzte wieder intensivere Forschungsarbeit im Bereich der neuronalen Netze ein; zum einen motiviert durch ein psychologisches Interesse, weil sich zeigte, daß einige von Minsky und Papert bewiesenen Beschränkungen auch auf das menschliche Gehirn zutreffen und damit der Modellcharakter erhalten bleibt, zum zweiten, weil Netzwerke und Algorithmen entwickelt wurden, die für Perzeptrone unlösbare Probleme lösen konnten.

Heutzutage ist die Zahl der entwickelten Netze unüberschaubar, jedoch lassen sie sich grob nach den von ihnen benutzten Lernalgorithmen unterscheiden: Netze mit unüberwachtem, selbstorganisierendem Lernen, eingeführt von Kohonen 1984, und Netze mit überwachtem Lernen nach dem 1986 von mehreren Forschern gleichzeitig entwickelten *error-backpropagation*-Algorithmus (vgl. Kapitel 2.6.1).

In der Psychologie scheint nun der Konnektionismus endgültig zum Durchbruch zu kommen, und es wird diskutiert, ob ein Paradigmawechsel im Sinne Kuhns (1976) stattfindet, wie z.B. Schneider (1987) behauptet. Vor allem zwischen den Anhängern des aus der Künstlichen-Intelligenz-Forschung entspringenden regelbasierten Symbolverarbeitungsansatzes und den Konnektionisten werden erbitterte Schriftgefechte um die Vorherrschaft in der psychologischen Modellbildung geführt (z.B. Rumelhart, McClelland u.a. bzw. McClelland, Rumelhart u.a. 1986, Smolensky 1988 als Anwälte des Konnektionismus; Fodor & Pylyshyn 1988 als Gegner; Helm 1991, Clark & Lutz 1992 als Vermittler).

In ihrer Kritik bestreiten Fodor und Pylyshyn zwar nicht, daß das Gehirn aus Neuronen und Verbindungen zwischen ihnen besteht und daß Gedanken als Erregung von Neuronen implementiert sein können, sehen aber darin keine Relevanz für die Psychologie, da sich dort psychische Vorgänge durch regelbasierte Symbolverarbeitung hinreichend erklären



lassen. Smolensky unterscheidet zwischen regelbasiertem, in Sprache codiertem Kulturwissen und individuellem Wissen, zu dem auch motorische Fertigkeiten und Intuition gehören. Letztere sind nicht sprachlich codiert und nicht aus expliziten Regeln ableitbar. Für sie kann nur auf subsymbolischer, sprich konnektionistischer Ebene eine sinnvolle Psychologie betrieben werden.

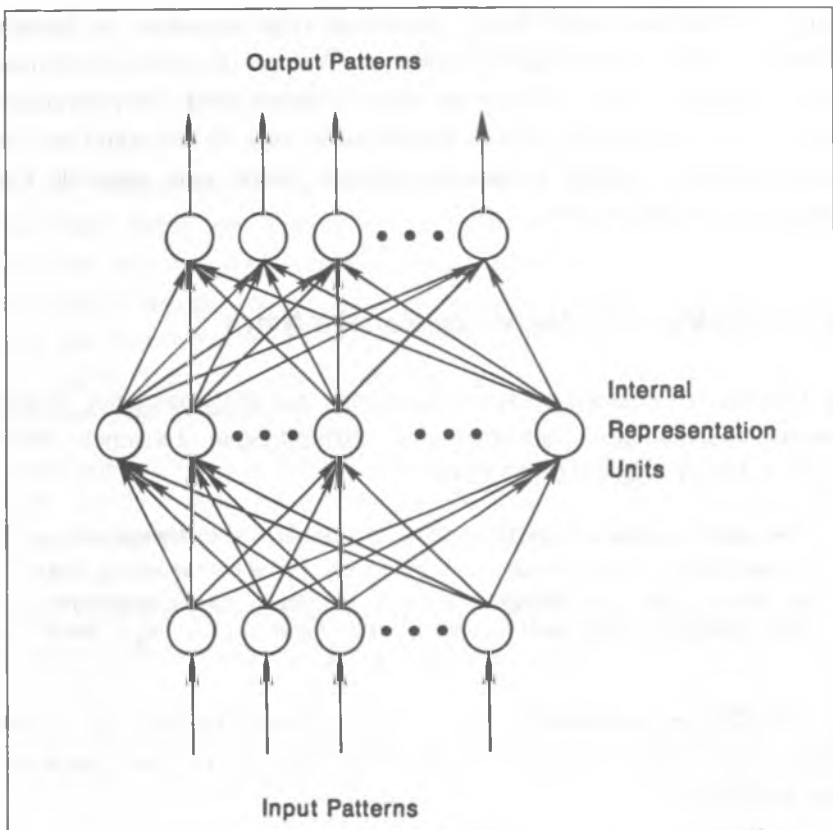
## 2.3 Architektur künstlicher neuronaler Netze

Alle künstlichen neuronalen Netze bestehen aus Eingabestellen, Ausgabestellen, künstlichen Neuronen und Verbindungen zwischen ihnen. Formal definiert Rojas (1993, 121):

Eine *Netzarchitektur* ist ein Tupel  $(E, N, A, T)$ , wobei  $E$  eine Menge von  $n_e$  Eingabestellen,  $N$  eine Menge von  $n$  Neuronen,  $A$  eine Menge von  $n_a$  Ausgabestellen und  $T$  eine Menge gewichteter gerichteter Kanten bezeichnet. Eine gewichtete gerichtete Kante ist ein Tupel  $(k_i, k_j, w_{ij})$ , wobei  $k_i \in E \cup N$ ;  $k_j \in N \cup A$  und  $w_{ij}$  eine reelle Zahl ist.

Je nach Zusammensetzung und Verbindungsmerkmalen kann man verschiedene Netze mit unterschiedlichen Lernalgorithmen gegeneinander abgrenzen.

*Mehrschichtige Netzwerke (multilayer networks)* gehören zu den gebräuchlichsten Netzarchitekturen. Das Netz besteht aus mehreren, zumeist drei Schichten. Die erste Schicht bezeichnet man als Eingabeschicht, die letzte als Ausgabeschicht, alle anderen Schichten sind verborgene Schichten. Alle Neuronen einer Schicht sind mit allen Neuronen einer benachbarten Schicht verbunden, weitere Verbindungen haben sie nicht. Die Verbindungen sind gerichtet, und zwar von der Eingabeschicht durch die verborgenen Schichten zur Ausgabeschicht. Da es keine Zyklen und keine Rückkopplung in diesen Netzen gibt, spricht man auch von vorwärtsgerichteten Netzen (*feedforward networks*). Ein solches Netz wird in Abbildung 4 dargestellt.



**Abb. 4: Architektur eines mehrschichtigen Netzwerks** (aus Rumelhart, Hinton & Williams 1986, 320).

Wenn man die Bedingung der Rückkopplungsfreiheit fallenläßt, erhält man *rekursive Netzarchitekturen*. Sie eignen sich besonders zur Verwendung als inhaltsadressierbare Speicher (Rojas 1993, 252). Im Gegensatz zu den mehrschichtigen Netzwerken, bei denen zumeist die Umgebung einer bekannten Eingabe in die Umgebung der jeweiligen Ausgabe abgebildet wird, wird bei kontextadressierbaren Speichern die Umgebung der bekannten Eingabe auf genau eine einzige Ausgabe abgebildet. Anschaulich gesprochen wird für verschiedene, aber ähnliche Eingaben, z.B. die verschiedenen Bewegungsmuster eines Flop-

Sprungs, derselbe Ausgabevektor generiert, wobei dann der Ausgabevektor die Bedeutung „Flop“ hätte. So können diese Netze zum Beispiel zur Mustererkennung eingesetzt werden. Bei rekursiven Netzen kann nicht mehr von einer Schichtarchitektur gesprochen werden, da durch die Rückkopplung die Neuronen nicht mehr eindeutig einzelnen Schichten zuzuordnen sind.

Im Extremfall sind alle Neuronen untereinander in beide Richtungen verbunden, eine Netzarchitektur, die nach ihrem Entwickler Hopfield-Modell genannt wird (Hopfield 1982).

## 2.4 Lokale Informationsverarbeitung

Die Lokale Informationsverarbeitung eines einzelnen Neurons folgt im wesentlichen denselben Prinzipien, die schon McCulloch und Pitts für ihre Zellen formuliert haben (s. Kapitel 2.2). Fünf Eigenschaften legen die Funktion von McCulloch-Pitts-Zellen fest:

- Eine McCulloch-Pitts-Zelle hat genau zwei mögliche Zustände, sie ist entweder erregt oder nicht erregt.
- Sie hat einen festen Schwellenwert.
- Sie erhält Eingaben von erregenden Synapsen.
- Sie kann auch Eingaben von hemmenden Synapsen erhalten. Diese Hemmung ist absolut, ist eine hemmende Synapse aktiv, ist das Neuron nicht erregt.
- Es gibt einen festgelegten Zeittakt für die Verarbeitung der Eingaben.

Abbildung 5 verdeutlicht den Aufbau einer McCulloch-Pitts-Zelle. Eine McCulloch-Pitts-Zelle ist erregt, wenn die Summe der Eingänge gleich oder größer ist als der Schwellenwert und kein hemmendes Neuron aktiv ist.

Als Erweiterung dieser Idee sind die Eingänge einer heute aktuellen Zelle gewichtet. Die eingehende Erregung wird dann mit einem Gewichtungsfaktor multipliziert. Auch wird die absolute Hemmung durch eine relative

ersetzt. Eine relative Hemmung kann im mathematischen Modell wie eine negative Gewichtung (oder eine negative Erregung) behandelt werden. Die so erhaltene Erregungsfunktion wird wegen ihrer Form auch Stufenfunktion genannt, wie aus dem Graph der Funktion deutlich wird (s. Abb. 6).

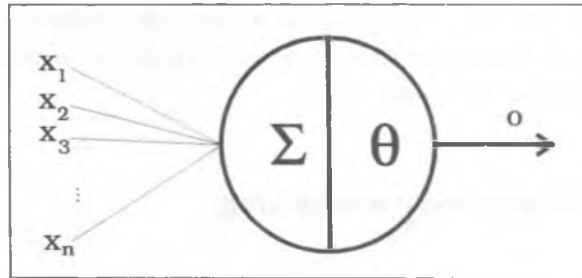


Abb. 5: Schematische Darstellung einer McCulloch-Pitts-Zelle.  $x_1$  bis  $x_n$  sind die Komponenten des Eingabevektors,  $o$  ist die Ausgabe und  $\theta$  der Schwellenwert.<sup>3</sup>

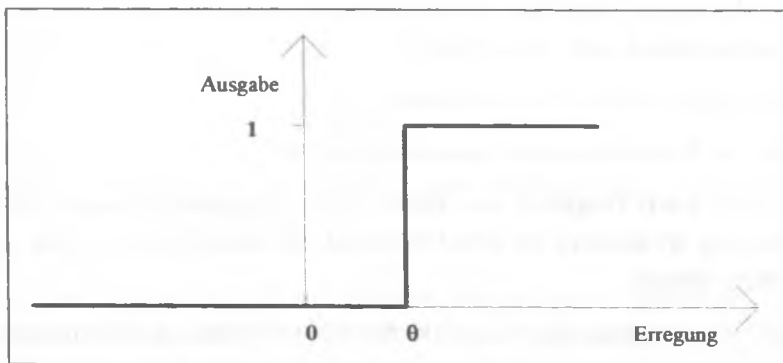


Abb. 6: Erregungsfunktion einer McCulloch-Pitts-Zelle.

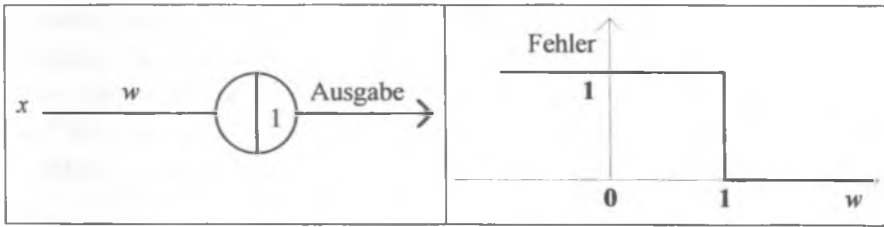
Der Schwellenwert kann auch als negative Erregung formuliert und so für alle Zellen der Schwellenwert Null festgesetzt werden, was lediglich

<sup>3</sup> Vektoren werden in Fettdruck dargestellt

die mathematische Behandlung komfortabler macht. Die mathematische Formel der Erregungsfunktion in diesem erweiterten Modell ist

$$\sum \mathbf{x}_i \mathbf{w}_i - \theta \geq 0 \quad (3)$$

Für eine festgelegte Ausgabefunktion kann über dem Gewichteraum eine Fehlerfunktion bestimmt werden. So ist z.B. für die Identitätsfunktion  $f(\mathbf{x}) = \mathbf{x}$  mit  $\mathbf{x}$  aus  $\{0,1\}$  in einer Zelle mit einer Eingabe- und einer Ausgabeleitung und dem festen Schwellenwert 1 der Graph der Fehlerfunktion in Abbildung 7 dargestellt.



**Abb. 7:** Graph der Fehlerfunktion der Identitätsfunktion bei einem Neuron mit dem Schwellenwert 1 (rechts die Funktion, links eine Darstellung des Neurons). Für  $w \geq 1$  entspricht die Ausgabe der Eingabe, d.h. die Identitätsfunktion wird korrekt abgebildet. Für  $w < 1$  ist der Fehler 1, da für  $x = 1$  das Netz den Wert 0 ausgibt.

Die Stufenfunktion wird mit Entwicklung des *backpropagation*-Algorithmus ersetzt durch eine stetige und ableitbare Annäherung, die Sigmoide (S-förmige).

$$s(x) = \frac{1}{1 + e^{-cx}} \quad (4)$$

Der Graph der Sigmoiden (Abb. 8) sieht aus wie die abgerundete Erregungsfunktion aus Abbildung 6, die der Grenzwert der Sigmoiden für  $c \rightarrow \infty$  ist. Im Gegensatz zu der Stufenfunktion ist die Sigmoide jedoch ableitbar. Ihre Ableitung ist

$$\frac{ds(x)}{dx} = \frac{ce^{-cx}}{(1 + e^{-cx})^2} = cs(x)(1 - s(x)) \quad (5)$$

Damit ist auch die entstehende Fehlerfunktion, die im drei- und höherdimensionalen Raum auch Fehlerlandschaft genannt wird, ableitbar.

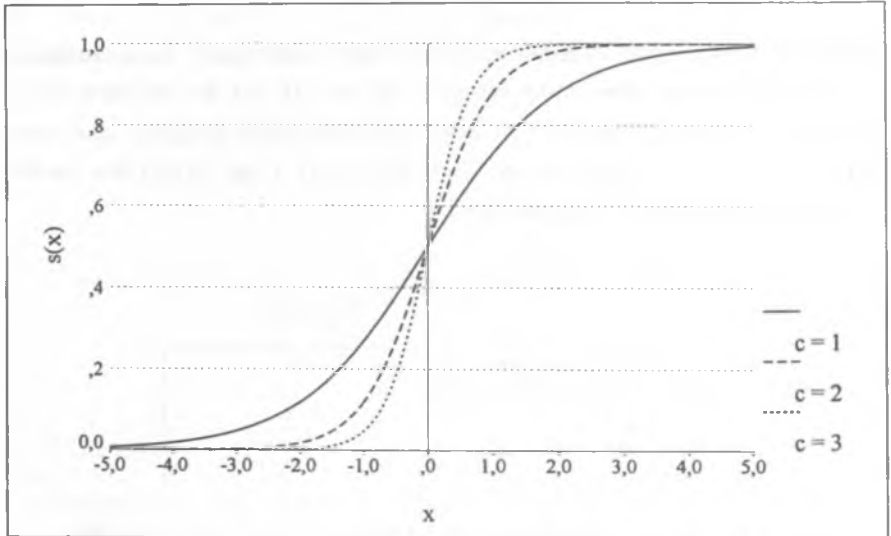


Abb. 8: Graph der Sigmoiden mit  $c = 1$ ,  $c = 2$  und  $c = 3$ .<sup>4</sup>

Grundsätzlich müssen von dieser Modellierung eines Neurons Neuronenmodelle unterschieden werden, die binäre Ausgabeerregungen besitzen, für die aber eine Übergangswahrscheinlichkeit definiert ist. Sie sind entweder erregt oder nicht erregt, und die Wahrscheinlichkeit für den jeweiligen Zustand ist abhängig von dem Input. Die Sigmoidfunktion aus Gleichung 4 ist eine häufig gewählte Funktion für die Übergangswahrscheinlichkeit.

<sup>4</sup> In Netzmodellen mit stochastischen Erregungsfunktionen gibt es Lernverfahren, bei denen metaphorisch Metallveredelungsverfahren simuliert werden. Dem langsamen Erkalten des Metalls entspricht das langsame Erniedrigen von  $c$ . Daher wird  $c$  auch oft als Temperaturkonstante bezeichnet.

## 2.5 Globale Informationsverarbeitung

Interessant im Zusammenhang mit Motorikforschung und kognitiver Modellbildung ist natürlich die globale Informationsverarbeitung. Wie reagiert das Netz, wenn man es als *ein* System betrachtet?

Alle Informationen, die gleichzeitig an den Eingabeneuronen anliegen, lassen sich als der Eingabevektor definieren, dementsprechend wird auch der Ausgabevektor festgelegt. Mathematisch gesehen vollführt das Netz nichts anderes als eine Vektortransformation von einem Eingabevektor in einen Ausgabevektor. Wie diese Transformation im Inneren bewerkstelligt wird, hängt ganz von der Netzarchitektur sowie von den Eigenschaften der lokalen Informationsverarbeitung ab. Trotzdem lassen sich allgemeine Eigenschaften der Netzwerke angeben, die sie für die kognitionspsychologische Modellbildung sehr interessant erscheinen lassen:

- Neuronale Netze speichern assoziativ, die Struktur des Eingabevektors bewirkt die Assoziation mit dem Ausgabevektor. Selbst wenn Informationen im Eingabevektor fehlen oder zuviel sind, wenn bei der Mustererkennung von Gesichtern beispielsweise im Gegensatz zu dem vorher trainierten Muster der Bart fehlt oder eine Brille hinzukommt, kann das Netz den richtigen Ausgabevektor liefern.
- Netze sind tolerant gegenüber zufälligen Eingabeschwankungen (*noise*). Verrauschte Eingaben sind bei menschlicher Mustererkennung eher die Regel als die Ausnahme.
- Künstliche neuronale Netze sind robust gegenüber dem Ausfall von einzelnen Neuronen. Sie „stürzen würdevoll ab“ („*degrade gracefully*“), was bedeutet, daß kein totaler Zusammenbruch des Netzwerks bei Ausfall einzelner Neuronen stattfindet.
- Neuronale Netze können lernen (s. Kapitel 2.6.).
- Bestimmte Fähigkeiten, die der Mensch leicht ausführen kann, wie das Wahrnehmen von Gesichtern oder im allgemeinen von optischen Mustern, gelingen auch künstlichen neuronalen Netzen recht gut. Sie

können, im Gegensatz zu herkömmlichen Computermodellen, zwischen Bäumen und Telegrafmasten (bzw. zwischen deren digitalisierten Abbildern) unterscheiden. Formale logische Operationen fallen ihnen aber recht schwer, was jedoch durchaus, wie beispielsweise beim Rechnen, seine Analogie im menschlichen Verhalten hat.

- Das Prinzip der parallelen, verteilten Informationsverarbeitung ist biologisch plausibler als die Konstrukte alternativer Modelle.

## 2.6 Lernen

Eine der wichtigsten Eigenschaften neuronaler Netze, vor allem auch für die kognitive Modellbildung, ist die Lernfähigkeit. Die Menge der verschiedenen Lernalgorithmen läßt sich in zwei Klassen einteilen, in überwachtes Lernen und unüberwachtes Lernen (Rojas 1993, 75).

### 2.6.1 Überwachtes Lernen

Beim überwachten Lernen stehen dem Netz Trainingsvektoren zur Verfügung, für die der korrekte Ausgabevektor einem „Lehrer“ bekannt ist. Der vom Netz produzierte Ausgabevektor wird mit dem korrekten Ergebnisvektor verglichen und der Fehler dem Netz zurückgemeldet. Daraufhin werden in dem Netz nach einem Algorithmus die Gewichte zwischen den einzelnen Neuronen verändert, um den Fehler zu minimieren. Nach Abschluß der Lernphase kann das Netz innerhalb gewisser Fehlertoleranzen jedem Trainingsvektor den dazugehörigen Ausgabevektor zuordnen. Die eigentliche Leistung besteht nun darin, daß das Netz auch unbekannten Eingabevektoren, die nicht zur Trainingsmenge gehören, Ausgabevektoren zuordnet. Wenn nichtlineare Aktivierungsfunktionen für die einzelnen Neuronen gewählt wurden, ist das Netz in der Lage, sehr komplexe Zusammenhänge zwischen den Trainingsvektoren und den jeweiligen Ausgabevektoren zu verallgemeinern.

Der zur Zeit wichtigste und effektivste Algorithmus für überwachtes Lernen wurde unabhängig voneinander von Parker (1985), Le Cun



(1986) und von der Forschergruppe Rumelhart, Hinton und Williams (1986) entwickelt. Letztere gaben dem Verfahren auch den heute üblichen Namen, *error-backpropagation*, zu deutsch „Fehlerrückführung“ oder „Fehlerfortschreibung“. *Error-backpropagation*, kurz *backpropagation*, ist ein Algorithmus für überwachtes Lernen. Er wurde für mehrschichtige Netze entwickelt. Wie schon in Kapitel 2.4 erläutert, ist bei einer sigmoidalen Aktivierungsfunktion die entstehende Fehlerfunktion ableitbar, das heißt in jedem Punkt ist die Steigung definiert. Interessanterweise erhält man, wenn man alle Verbindungen in einem Netz von der Ausgabeseite mit dem Fehler, den es produziert hat, durchläuft und statt der Sigmoiden als Aktivierungsfunktion ihre Ableitung nimmt, als Ausgabe eines jeden Neurons die partielle Ableitung der Fehlerfunktion über die durchlaufene Kante. Nun braucht man nur noch um eine Konstante  $\gamma$  in Richtung des negativen Gradienten das durchlaufene Gewicht zu ändern, und in der Regel wird der Fehler sich verringern.

Geometrisch interpretiert entspricht der Zustand des Netzes einem Punkt auf der dem System unbekannten multidimensionalen Fehlerlandschaft. Für diesen Punkt ist durch den Vergleich mit dem Zielvektor der Fehler bekannt, durch das Rückführen des Fehlers auch die Steigung. Das Verändern der Gewichte in Richtung des negativen partiellen Gradienten kann als Talwärtsrollen (oder besser -springen) einer Kugel auf der Fehlerlandschaft anschaulich gemacht werden, wobei der Ausgangspunkt der aktuelle Netzzustand ist, der Zielpunkt ein Tal mit dem Fehler Null. Durch dieses Bild werden auch die Probleme des *backpropagation*-Algorithmus' deutlich: Wählt man die Konstante  $\gamma$  zu groß, kann das Null-Fehler-Tal „übersprungen“ werden, bei einem zu kleinen  $\gamma$  besteht andererseits die Gefahr, in eventuell existierenden Nebentälern (lokalen Minima) „hängenzubleiben“. Über die Verfahren, diesen Schwierigkeiten zu begegnen und für ein genaueres Verständnis des Algorithmus' sei auf die Fachliteratur (vor allem Rojas 1993, 149 ff) verwiesen.

Durch die Entdeckung dieses Lernverfahrens wurden die Grenzen von Rosenblatts Perceptron-Lernen (s. Kapitel 2.2) aufgehoben und damit auch die Konstruktion größerer Netze mit mehreren Schichten ermöglicht.

### 2.6.2 Unüberwachtes Lernen

In vielen Anwendungsfällen ist jedoch der zu einer Eingabe gehörende Ausgabevektor unbekannt. Prominentestes Beispiel dafür ist die Mustererkennung. Hier soll das Diskriminieren verschiedener Muster gelernt werden, wobei es eine Frage der Aufmerksamkeit ist, ob es sich bei zwei Mustern nun um verschiedene oder um gleiche Kategorien handelt. Unter statistischem Blickwinkel ist die Aufgabe des Netzes eine Clusterbildung, wobei sich oft jeweils ein Neuron der Ausgabeschicht auf ein Cluster spezialisiert.

Ein typisches Lernverfahren besteht darin, die Erregung der Neuronen in der Ausgabeschicht von dem Abstand des Gewichtevektors zum Eingabevektor abhängig zu machen. Je dichter diese beiden Vektoren nebeneinander liegen, desto mehr wird das zum Gewichtevektor gehörige Ausgabeneuron erregt. Ein *winner-takes-all* Algorithmus in der Ausgabeschicht sorgt dafür, daß das am meisten erregte Neuron feuert und gleichzeitig alle anderen Neuronen der Ausgabeschicht gehemmt werden. Der zum Ausgabeneuron gehörende Gewichtevektor wird so modifiziert, daß er etwas näher hin zum Eingabevektor rückt. Bilden mehrere Eingabevektoren ein Cluster, wird dadurch mit der Zeit der Gewichtevektor in ihrer Mitte plaziert. Die „Breite“ des „Clusterkegels“ wird durch einen „Aufmerksamkeitsparameter“ global gesteuert. Welches Neuron auf welches Cluster anspricht, steuert sich „selbstorganisierend“ und kann nicht vorhergesehen werden. Auch hierbei gilt, daß bei nichtlinearen Aktivierungsfunktionen sehr komplexe Gemeinsamkeiten zwischen den einem Cluster zugehörenden Vektoren durch das Netz gefunden werden können.

### 2.7 Interne Repräsentationen

In der bewegungswissenschaftlichen Auseinandersetzung der verschiedenen Ansätze, vor allem der „*motor-action* Kontroverse“, d. h. der Kontroverse zwischen Anhängern des Programm- und des psychoökologischen Ansatzes, spielt der Begriff der internen (oder zentralen)

Repräsentation eine bedeutende Rolle (vgl. Meijer & Roth 1988). An dieser Stelle muß eine kurze Bestimmung des Begriffs der „Repräsentation“ eingeschoben werden, denn „ ‚Repräsentation‘ ist notorisch vieldeutig, so sehr, daß seine ‚unerläuterte‘ Verwendung eine ‚geistige Gänsehaut‘ (Kämmerling 1989) erzeugen mag“ (Scheerer 1993, 137).

Nach Scheerer (a.a.O.) gibt es bei historischer Betrachtung vier Gruppen von Verwendungsweisen.

- Der weiteste Repräsentationsbegriff stammt von Hobbes. Repräsentation bedeutet hier Stellvertretung, wenn eine Sache oder Person für irgendeine andere Sache oder Person steht, gleichgültig, ob zwischen beiden Ähnlichkeit besteht oder nicht.
- Auf Descartes' Ideenlehre zurückgehend bedeutet Repräsentation jede Art von mentalem Inhalt, aber auch jede Art von inhaltsvollem mentalen Akt und ist ebenfalls eine sehr weite Fassung des Repräsentationsbegriffs.
- Repräsentation kann auch im Sinne einer bildlichen oder strukturerhaltenden Darstellung definiert werden. Diese Begriffsbestimmung wird schon seit der Antike verwendet und ist nicht nur auf mentale Repräsentationen beschränkt. Häufig wird jedoch angenommen, daß mentale Repräsentationen bildhafte Ähnlichkeiten zu dem Repräsentandum aufweisen müssen. Strukturerhaltende Darstellungen können jedoch auch isomorphe Abbildungen im mathematischen Sinn sein, wie z.B. eine geometrische und eine algebraische Darstellung ein und desselben Sachverhalts in der analytischen Geometrie. Dieser Bedeutungszusammenhang wurde von Leibniz eingeführt.
- Eine Einschränkung des Definitionsbereichs von Repräsentation umfaßt den Gesamtbereich der vermittelten Erkenntnis, also Gedächtnis und Denken, schließt aber die Wahrnehmung als sinnliche Erkenntnis aus. Sie geht auf Hegel zurück und setzt sich um 1900 in der Fachsprache durch.

In der Kognitionspsychologie wurde der Begriff der Repräsentation von der Linguistik und der KI-Forschung übernommen. Jedoch war

„spätestens im Jahre 1975 (...) das Konzept der mentalen (oder ‚inneren‘) Repräsentation in jenen Disziplinen fest verankert, die heute zur Kognitionswissenschaft gehören“ (Scheerer 1993, 145). Dort wird der Begriff eingebunden in die *Language of Thought* (Fodor 1975). Diese „Sprache des Geistes“ ist eng an die KI-Forschung angelehnt. Repräsentationen sind darin innere Zustände eines Symbolsystems, die formal interpretiert werden. Wegen der kontextfreien Semantik entspricht die semantische Interpretation der formalen. Symbole und damit auch Repräsentationen sind inhaltlich logisch, das heißt Träger von Wahrheitswerten und syntaktisch sentential, also „Sätze“ mit mindestens einem Subjekt und einem Prädikat.

Gegen diesen Repräsentationsbegriff setzt sich der des Konnektionismus' scharf ab. Es gibt zwei Auffassungen, wie in neuronalen Netzwerken repräsentiert wird. „Wissen steckt in den Gewichten“ (vgl. Smolensky 1988, 1) besagt, daß alle Repräsentationen verteilt in den Gewichtematrizen der Verbindungen zu den verdeckten Einheiten stecken; oder der Repräsentation entspricht „das durch die jeweilige Eingabeinformation auf dem Hintergrund eines Konnektivitätsmusters erzeugte Aktivierungsmuster der verdeckten Einheiten“ (Scheerer 1993, 158). Beide Sichtweisen schließen sich nicht aus; während in der Gewichtematrix das gesamte Wissen verteilt repräsentiert ist, entspricht dem Aktivitätsmuster eine konkrete Repräsentation, die während der Anwendung entsteht. Wie schon erwähnt, ist eben nicht eine einzelne Aktivität eines Neurons für die Repräsentation verantwortlich, sondern sie ergibt sich aus dem *Muster* der Aktivitäten aller Einheiten der verdeckten Schicht. Deswegen spricht man von *verteilter* Repräsentation. Diese Muster sind *emergent*, sie entstehen für jeden Eingabevektor neu. Konnektionistische Repräsentationen bilden sich selbstorganisierend heraus, dadurch werden Repräsentieren und Lernen ein einheitlicher Prozeß. Die Bedeutung der Repräsentation kann daher nicht innerhalb des Netzes bekannt sein, sondern ist durch die Systemumgebung determiniert. „Was ein solches Muster bedeutet, ergibt sich nicht aus einer systeminternen Semantik, sondern nur aus der *Interaktion mit der Umwelt*“ (Scheerer 1993, 160, Hervorh. d. Verf.).

Wie schon in Kapitel 2.3 erwähnt, werden in neuronalen Netzen benachbarte Eingabeumgebungen auf benachbarte oder gleiche Ausgabeumgebungen abgebildet. Dies bedeutet in dem Zusammenhang dieses Kapitels, daß für ähnliche Gegenstände ähnliche Repräsentationen instantiiert werden. Damit ist also die Repräsentation im Konnektionismus eine isomorphe Abbildung.



### **3 Konnektionistische Modelle motorischer Kontrolle und motorischen Lernens**

Das Feld der motorischen Kontrolle und des motorischen Lernens war in den frühen Jahren der modernen konnektionistischen Forschung kein Gegenstand des Erkenntnisinteresses. Von zwei Ausnahmen ist zu berichten: Rumelhart und Norman (1982) untersuchten die Fingerbewegungen von professionellen Büroangestellten beim Schreibmaschinenschreiben. Dabei entdeckten sie, daß die Fingerbewegung zum Anschlag einer Taste nicht von je einem Programm pro Taste gesteuert wird, sondern daß je nach Buchstabenkontext auch die Fingerbewegung für einen einzelnen Anschlag unterschiedlich war. Es gab also unterschiedliche Bewegungsmuster für das Niederdrücken einer Taste in Abhängigkeit von den vorausgehenden und folgenden Buchstaben. Dieses Verhalten modellierten sie in einem künstlichen neuronalen Netzwerk. Mit einem ähnlichen Netz modellierte Hinton (1984) ein Strichneutrum, das einen Gegenstand aufheben sollte, ohne dabei das Gleichgewicht zu verlieren.

Diese frühen Modelle zur motorischen Kontrolle wurden aber ohne einen Bezug zu bewegungswissenschaftlichen oder psychomotorischen Theorien erstellt. Erst in jüngster Zeit haben sich Modelle entwickelt, die motorische und konnektionistische Theorien in Zusammenschau betrachten. Unter diesen Modellen sind zwei, die bisher mit Abstand am häufigsten Eingang in die bewegungswissenschaftliche Diskussion gefunden haben, das VITE-Modell von Bullock und Grossberg und das Netzwerk von Jordan und Kollegen. Diese beiden Modelle sollen ausführlich erläutert werden.

#### **3.1 Das VITE-Modell von Bullock und Grossberg**

Stephen Grossberg, ein amerikanischer Biologe und Mathematiker, ist einer der fleißigsten Produzenten neuronaler Netzwerke, der sich auch in den *dark ages* (s. S. 38) seine Forschungsarbeiten daran nicht verdrie-

ßen ließ. Mitte der 80er Jahre entwickelte er zusammen mit Kuperstein Modelle zur Koordination und Kontrolle von Augenbewegungen (Grossberg & Kuperstein 1986). Die dort gewonnenen Erkenntnisse übertrug er mit Bullock auf die Steuerung von Greifbewegungen. Ab 1988 präsentierten die beiden Forscher in zahlreichen Veröffentlichungen (Bullock & Grossberg 1988a, 1988b, 1988c, 1989, 1990, 1991a) ihr VITE-Modell, das bisher in der Bewegungswissenschaft meistzitierte neuronale Netz (z.B. das Februarheft 1991 von Human Movement Science nur zu diesem Thema; Rosenbaum 1991, 382-386; Morasso & Sanguineti 1992, 13-15).

Eine Einordnung in den bewegungswissenschaftlichen Forschungszusammenhang nehmen Bullock und Grossberg (1988a, 50) selbst vor:

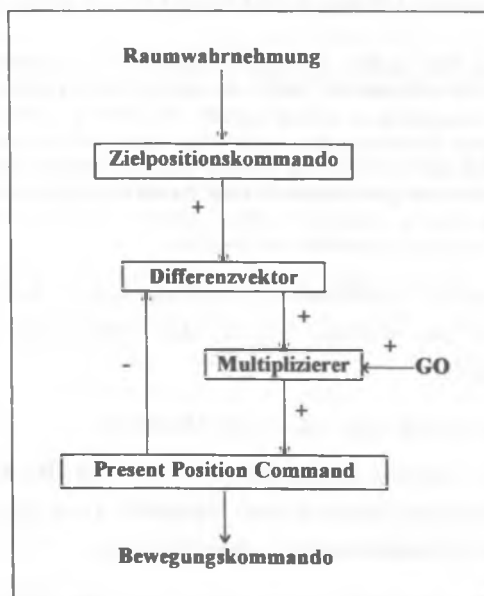
Our work shows that neither an explicit trajectory nor a kinematic invariant need be explicitly represented within a motor-control system at any time. Thus our work supports a critical insight of workers in the mass-spring modeling tradition that movement kinematics need not be explicitly preprogrammed. By the same token, our results reject a mass-spring model in its customary form and argue against models based on optimization theory. Instead we show how a movement-control system may be adaptive without necessarily optimizing an explicit cost function .

Bullock und Grossberg modellieren ein *inflow signal*, die Afferenzen oder Reafferenzen und ein *outflow signal*, die Efferenzen. Die Efferenzen haben drei Funktionen:

- Sie senden Bewegungssignale an die Muskeln.
- Sie generieren *corollary discharges*, die mit den Reafferenzen verglichen werden können. Durch diesen Vergleich kann das Verhältnis von Efferenz und Muskelkontraktion gelernt werden.
- Durch denselben Vergleich wird gelernt, aus den Bewegungssignalen an die Muskeln die aktuelle Position zu schätzen (Efferenzkopie).

Die Reafferenzen sind nötig, um die aktuelle Position zurückzumelden, wenn dies auf efferentem Wege noch nicht gelernt oder nicht möglich ist, wie bei passiven Bewegungen. Das VITE-Modell geht davon aus, daß die Abbildung der Efferenzkopie auf die aktuelle Position schon gelernt wurde.

Abbildung 9 erläutert die Funktionsweise des VITE-Modells. Eine Bewegung wird hauptsächlich durch zwei Kommandos, dem Aktuelle-Position-Kommando (present position command, *PPC*) und dem Zielpositionskommando (target position command, *TPC*) gesteuert. Dabei sind *TPC* und *PPC* Vektoren im Muskelraum, die spezifizieren Muskellängen, die der jeweiligen Armposition zugeordnet sind. Das *TPC* kann durch eine gelernte assoziative Verbindung zum Sehzentrum festgelegt werden, das *PPC* wird aus der Efferenzkopie geschätzt. Es handelt sich bei dem VITE-Modell also um eine *open-loop*-Steuerung, die nicht auf Rückmeldung angewiesen ist.



**Abb. 9: Das VITE-Modell** (nach Bullock & Grossberg 1988a, 71). Die einzelnen Rechtecke stehen für Schichten von Neuronen, deren Funktion durch die Aufschrift erklärt wird (nähere Erläuterungen im Text).

Ist der Arm in Ruhe, sind *TPC* und *PPC* in der Regel gleich. Der Wert des Differenzvektors *DV*, der zu jeder Zeit im Netz entsteht, ist Null. Wenn jetzt eine Bewegung ausgeführt werden soll, wird das Zielposi-



tionskommando  $TPC$  neu spezifiziert. Das  $PPC$  ändert sich nun nach folgender Gleichung:

$$\frac{dPPC_i}{dt} = GO(t) \cdot [DV_i(t)]^+ \quad (6)$$

dabei sei  $i$  ein einzelner von  $n$  spezifizierten Muskeln und

$$[DV]^+ = \begin{cases} DV & \text{für } DV > 0 \\ 0 & \text{für } DV \leq 0 \end{cases} \quad (7)$$

in Worten: die Änderung des  $PPC$  für einen Muskel erfolgt in Richtung und proportional zur Größe von  $DV$  und einem  $GO$ -Signal, wenn der Differenzvektor positiv ist, wenn es sich also um einen synergistisch wirkenden Muskel handelt. Ein Antagonist erfährt keine Änderung des  $PPC$ , da für ihn Gleichung 6 Null wird. Das  $GO$ -Signal modelliert die willkürliche Entscheidung über Bewegungsbeginn sowie Bewegungsgeschwindigkeit und ist global und gleich für alle Muskeln.  $PPC$  integriert also  $DV$  multipliziert mit einem willkürlichen Signal, daher auch der Name des Modells, „Vektor / Integration To Endpoint“, oder VITE.  $DV$  hängt allerdings auch kontinuierlich von der Zeit ab, die nachstehende Gleichung erläutert diesen Zusammenhang:

$$\frac{dDV_i}{dt} = \alpha (-DV_i(t) + TPC_i - PPC_i(t)) \quad (8)$$

Die Änderung des Differenzvektors hängt ab von dem (zeitlich konstanten) Ziel und der sich kontinuierlich ändernden aktuellen Position und deren Differenz zur Zielposition. Dabei ist die Änderung von  $DV$  um so größer, je weiter Ziel und aktuelle Position auseinander liegen ( $TPC - PPC$ ) und je größer der Wert von  $DV$  ist, was eine Art „Schwung“ in die Sache bringt. Für ein zeitlich konstantes  $GO$ -Signal wurden die Differentialgleichungen von Bullock und Grossberg analytisch gelöst (1988a, 87-88), mit einer sigmoidalen Funktion (vgl. S. 44) für  $GO(t)$  führten sie Computersimulationen durch. Dies führte zu Ergebnissen, die eine Reihe von empirischen Daten der menschlichen Bewegungskontrolle durchaus adäquat wiedergeben können:

- Das „Einfrieren“ einer Bewegung während der Bewegungsdurchführung kann sehr leicht geschehen, indem das willkürliche *GO*-Signal auf Null gesetzt wird, dadurch wird in Gleichung 6 die Änderung des *PPC* Null und es findet keine Bewegung mehr statt.
- Das Phänomen, daß gezielte Handbewegungen auf einem geraden Weg durchgeführt werden (Morasso 1981), wird im Modell dadurch erreicht, daß die Änderung des *PPC* für jeden Muskel linear von *DV* abhängt (Gleichung 6). Muskeln mit größerem positiven *DV* kontrahieren also schneller als solche mit kleinem.
- Der *speed-accuracy trade-off*, der besagt, daß Bewegungsfehler proportional von der Distanz abhängen, und Fitts' Gesetz (Fitts 1954), daß die durchschnittliche Bewegungsdauer  $\bar{D}$  zwischen zwei Zielen mit ihrer Entfernung voneinander ( $E$ ) und der Breite der einzelnen Ziele ( $B$ ) logarithmisch nach der Formel  $\bar{D} = a + b \cdot \log(2E/B)$  zusammenhängen, können im VITE-Modell sowohl analytisch als auch experimentell nachgewiesen werden.
- Der Precuingeffekt (Rosenbaum 1983; Roth 1989) kann modelliert werden. Liegen vor der Bewegungsausführung schon Informationen über das Bewegungsziel vor, vermindert sich die Reaktionszeit. Im Modell kann dies dadurch erklärt werden, daß bei vorliegendem *TPC* der *DV* schon entstehen kann, auch wenn das *GO*-Signal noch Null ist. Wird dann willkürlich das *GO*-Signal hochgesetzt, ist *DV* schon korrekt, was die Reaktionszeit verkürzen kann.
- Die von Georgopoulos, Kalaska und Massey (1981) beobachteten Bewegungsbahnen bei Änderungen des Ziels während der Bewegungsausführung können in der Simulation vom Modell nachvollzogen werden.
- Die von Zelaznik, Schmidt und Gielen (1986) festgestellte Invarianz des Geschwindigkeitsprofils für verschiedene Ausführungsgeschwindigkeiten wurde auch bei Simulationen im Modell festgestellt.

### 3.1.1 VITE in der wissenschaftlichen Diskussion

Das VITE-Modell ist in der Lage, durch die geschickte Anordnung von Neuronenmodellen eine Reihe von Daten zu simulieren, die empirisch nachweisbar auch bei tierischen und menschlichen Greifbewegungen auftreten. Die meisten Forscher, die sich mit dem Netz von Bullock und Grossberg auseinandergesetzt haben, halten für die wichtigste Eigenschaft des Modells, daß das Modellverhalten nicht explizit berechnet wird, sondern aus dem Zusammenspiel der Neuronenmodelle entsteht. Beek und Bootsma (1991, 89) schreiben:

We are impressed with the possibilities of the model provided by Bullock and Grossberg, both in terms of the amount of data explained without seeking recourse to intelligent operational procedures as well as in terms of the "openness" of its architecture, allowing critical testing and future modification.

Kelso (1991, 99) lobt:

a nice feature of the VITE model is that kinematic invariants emerge through network interactions rather than an explicitly precomputed command.

Morasso hebt eine weitere wichtige Eigenschaft hervor, die für Modellierer von Netzwerken beeindruckend ist: „VITE has a *temporal competence*“ (1992, 15). Mit dem VITE-Modell lassen sich zeitliches Verhalten und Geschwindigkeitsverläufe simulieren. Diese Modelleigenschaft ist eher untypisch für konnektionistische Modelle. Sie ist Grossbergs Verdienst, der einer der wenigen Forscher ist, der Differentialgleichungen über die Zeit als Aktivierungsfunktionen für seine Neuronenmodelle benutzt (z.B. Grossberg 1982).

Bullock und Grossberg reklamieren für ihr Modell eine biologische Plausibilität. Sie sehen durch neurophysiologische Untersuchungen die Existenz ihrer Neuronenmodelle im lebenden Organismus bestätigt, die sie in einer weiteren Publikation (Bullock & Grossberg 1991a) veröffentlichen. Zusammenfassend schreiben sie: „We have reviewed evidence that T [TPC] is computed in parietal cortex, V [DV] in motor cortex, and G [GO-Signal] in globus pallidus“ (Bullock & Grossberg 1991b, 135).

Jedoch können auch die Beschränkungen des Modells nicht übersehen werden. So entsprechen nicht alle Simulationen im Modell exakt dem menschlichen Bewegungsverhalten. Bei genauer analytischer Betrachtung der Gleichungssysteme wird erkennbar, daß bei der Simulation des *speed-accuracy trade-off* VITE immer um einen konstanten Betrag über das Ziel hinausschießt. In Laboruntersuchungen wurde gezeigt, daß menschliche Versuchspersonen mit einer gewissen Abweichung um das Ziel streuen. Heuer (1991, 61) schlägt hier vor, in das Modell Zufallsvariable einzubeziehen. Bei der Simulation von geradlinigen Handbewegungen merkt Heuer (a.a.O., 60) an, daß durch die Transformation aus dem Muskellängenvektorraum, in dem VITE arbeitet, in den äußeren Raum die Geradlinigkeit verloren gehen muß. Unklar bleibt auch, was nach Bewegungsende mit dem *GO*-Signal passiert. Im Modell wird dies nicht weiter spezifiziert, obwohl dies Auswirkungen auf nachfolgende Bewegungen haben muß. Dazu bemerkt Deno (1991, 76): „This untidiness leaves the entire scheme suspect for composite or repetitive motions.“

Diese oben erwähnten kleinen Fehler innerhalb des Modells wiegen jedoch nicht schwer, da sie durch einfache Modifikationen behebbar erscheinen. Die in der bisher erschienenen Literatur zu VITE formulierte Kritik kann unter drei Überschriften zusammengefaßt werden:

- Das Modell ist zu simpel.
- Die Art der Informationsaufnahme ist falsch modelliert.
- Die Analyseebene ist falsch gewählt.

#### Das Modell ist zu simpel

Diese Kritik wird vor allem aus einem biomechanischen oder neurophysiologischen Blickwinkel geäußert. Das biomechanische Modell zu VITE enthält ein einziges Gelenk, das Bewegungen in einer Ebene zuläßt. Folglich werden keine Muskeln oder Bänder modelliert, die über mehrere Gelenke hinweg arbeiten. Gravitation und Elastizität werden im Modell ebensowenig berücksichtigt wie Zentrifugal- und Corioliskräfte. Alle spielen aber auch schon bei Greifbewegungen eine wichtige Rolle (Deno

1991, 75). Es stellt sich die berechnigte Frage, ob das Modell unter Berücksichtigung dieser komplexen Zusammenhänge menschliches oder tierisches Bewegungsverhalten noch adäquat beschreiben kann.

Aus neurophysiologischer Sicht werden ähnliche Bedenken geäußert. Das Zusammenspiel von Muskelspindeln,  $\alpha$ - und  $\gamma$ -Motoneuronen, Renshaw- und Purkinjezellen, Interneuronen, dem zentralen Nervensystem und weiteren entdeckten und unentdeckten Zellen und Molekülen bei der Ausführung von Bewegungen ist hoch komplex und physiologisch noch nicht geklärt. Daß die von Bullock und Grossberg postulierten Neuronenverbände und ihre Interaktionen physiologisch plausibel sind, kann daher nicht angenommen werden, wie Wilberg (1991) nachweist.

#### Die Art der Informationsaufnahme ist falsch modelliert

Diese Kritik wird hauptsächlich von Vertretern psychoökologischer Theorien geäußert. Ihnen erscheint unplausibel, daß räumliche Informationen aufgenommen werden, sie bevorzugen Modelle mit räumlich-zeitlicher Wahrnehmung. „Interestingly enough, in many settings temporal information seems to have primacy over positional information“ (Beek & Bootsma 1991, 89). So postulieren sie die Wahrnehmung einer Variablen, die die Zeit bis zum Auftreffen eines sich nähernden Gegenstandes auf die Gesichtsebene repräsentiert, oder die Wahrnehmung eines *optical flow fields* (vgl. auch Kap. 1.1.2). Diese Art von Wahrnehmungsleistung kann von VITE natürlich nicht simuliert werden.

#### Die Analyseebene ist falsch gewählt

Ansatzpunkt der Kritik von Wissenschaftlern, die sich mit synergetischer Modellbildung beschäftigen, an VITE ist, daß auf der Ebene von Neuronen oder Neuronenverbänden Erklärungen des Bewegungsverhaltens nicht möglich sind (Haken 1991). In der synergetischen Denkweise liegen dem beobachtbaren Verhalten einige wenige Ordnungsparameter auf höherer Ebene, wie z.B. Energieverbrauch oder Bewegungsfrequenzen, zugrunde.

### 3.1.2 Das Problem der Freiheitsgrade

Das VITE-Modell hat einen hochinteressanten Ansatz zur Lösung des Problems der Freiheitsgrade. Es scheint so zu sein, daß das *GO*-Signal der einzige willkürlich zu regulierende Freiheitsgrad bei der Greifbewegung ist. Aufwendige Optimierungsverfahren, wie sie in anderen Modellen nötig sind, entfallen. Doch bei genauer Betrachtung erscheint das Problem der Freiheitsgrade nur verschoben. Am Beispiel der Greifbewegung kann dies deutlich illustriert werden. Wenn man eine Tasse greifen will, die in Reichweite auf einem Tisch steht, so wird eine einfache Armbewegung genügen, um sie zu ergreifen. Die hierbei kontrollierten Gelenke sind neben den Hand- und Fingergelenken das Schulter- und das Ellbogengelenk, die aktive Muskulatur die Schulter- und Armmuskeln. Das Ziel der Greifbewegung, die Position der Tasse, ließe sich also als eine Funktion von Arm- und Schultermuskulatur darstellen. Dementsprechend wären die spezifizierten *TPCs* Vektoren der Längen der Arm- und Schultermuskeln. Was passiert jedoch, wenn die Tasse sich nicht mehr in Reichweite befindet? Üblicherweise neigt der Mensch dann den Oberkörper etwas nach vorn, während gleichzeitig die Armbewegung stattfindet. In Bullock und Grossbergs Modell müßte nun die gesamte Muskulatur, die beim Vorbeugen des Oberkörpers aktiviert wird, zur Spezifikation von *TPC* beitragen. Dieses Beispiel läßt sich noch weiter auf die Spitze treiben. Reicht auch das Vorlehnen nicht aus, um einen Gegenstand zu erreichen, wird der Mensch vom Stuhl aufstehen, sollte dieser immer noch unerreichbar sein, einen Schritt nach vorne tun. So tragen immer mehr Muskeln zur Greifbewegung bei, je größer die Entfernung von Körper und Gegenstand ist.

Da jedoch kein Mechanismus in dem Modell vorhanden ist, der die Entfernung eines Gegenstands registriert, müssen konsequenterweise die Muskellängen der gesamten Skelettmuskulatur zur Spezifikation von *TPC* herangezogen werden. Dies klingt aber nicht plausibel, zumal das Problem der Freiheitsgrade dann bei der zu lernenden Abbildung von Muskellängen zu Zielpositionen neu entsteht.

Zur Rettung des Modells kann argumentiert werden, daß für das Beugen

des Oberkörpers nach vorn und die Greifbewegung des Arms zwei unabhängige VITE-Netze zuständig sind. Dies mag für Bewegungen, die relativ unabhängig voneinander sind und sequentiell nacheinander ausgeführt werden, stimmen. Beim Vorbeugen und Greifen trifft dies jedoch nicht zu, wie der Leser und die Leserin leicht selbst am Schreibtisch nachvollziehen können.

### 3.1.3 FLETE und AVITE

Bullock und Grossberg ergänzen 1991 ihr Modell, um verschiedene Muskelspannungszustände im Gleichgewichtszustand simulieren zu können. Wenn ein Arm durch ein externes Gewicht belastet wird, ändern sich die Spannungszustände der Muskulatur. Um eine bestimmte Position zu erreichen, muß ein anderer Gleichgewichtspunkt angesteuert werden als ohne Last. Diesem Problem begegnen Bullock und Grossberg mit FLETE, der „Factorization of Length and Tension“ (Bullock & Grossberg 1991a, 28). Ein weiteres unspezifisches globales Signal, ähnlich dem *GO*-Signal, dient der *compliance control*, der Kontrolle der Muskelspannung. In einem an VITE angeschlossenen Netzwerk wird der komplizierte Zusammenhang zwischen Muskellänge, -anspannung und dem die Position bestimmenden Gleichgewichtszustand hergestellt. Über die Bedeutung des FLETE-Moduls für die Bewegungssteuerung können nur sehr begrenzte Aussagen gemacht werden. Deno stellt fest: „The FLETE section is difficult to read because of its disordered presentation of symbols and ideas“ (Deno 1991, 74), und Heuer beschäftigt sich nur mit dem VITE-Modell: „First, this model refers to a higher level of motor control than FLETE and is thus nearer to the mind of a psychologist; second, it is mathematically simpler“ (Heuer 1991, 55).

Gaudio und Grossberg (1992) befassen sich mit dem Lernen von VITE. Dazu erweitern sie dieses Modell zu AVITE, „Adaptive Vector Integration to Endpoint“. In der sogenannten Phase des motorischen Stammelns (*motor babbling phase*, Gaudio & Grossberg 1992, 142) werden die Abbildungen zwischen *TPC*, *PPC* und *DV* selbstorganisierend gelernt. Ein zufälliges Signal wird in *PPC* eingegeben und zu einem

Bewegungsbefehl integriert. Wenn die Bewegung stoppt, wird eine *now print*-Schleuse geöffnet, die Transformation des aktuellen *PPC* in die Ebene des *TPC* kopiert. *TPC* und *PPC* werden nun konstant gehalten, und die Gewichte zwischen *TPC* und *DV* werden so geändert, daß *DV* Null wird. Gaudiano und Grossberg berichten, daß in Computersimulationen eines eingelenkigen Arms mit Synergist und Antagonist der Lernalgorithmus den Gesamtfehler gegen Null konvergieren läßt.

### 3.1.4 Fazit

Bullock und Grossberg entwickeln mit VITE ein künstliches neuronales Netz zur Simulation von Greifbewegungen. Aus der Differenz von Zielposition und auf efferentem Wege bestimmter aktueller Position im Muskelvektorraum entsteht eine Bewegung, die ohne Berechnung von Optimierungsfunktionen oder Steuerungssignalen empirisch gefundenen Bewegungsmustern des Menschen ähnelt. Durch die dynamische Interaktion von Neuronen vereint das Modell gefundene Lösungen auf implementationeller Ebene (das Masse-Feder-Modell) mit Konstrukten der anderen Ebenen (unspezifische zentrale Signale, aufgabenbezogene Synergien). Dies fasziniert viele Forscher am VITE-Modell.

Die Attraktivität des Modells im Vergleich zu ähnlichen Modellen liegt aber auch darin begründet, daß es sich um ein künstliches neuronales Netz handelt und daher die postulierten Eigenschaften in einer Computersimulation überprüft werden können.

Although Bullock and Grossberg share a great many ideas (e.g., nonspecific central commands, muscle synergies) with others who have been engaged in the first round [of theorizing on Bernstein's problem], they go beyond much of the previous work to give mathematically precise interpretations of a number of vague notions that appealed to first round scholarship – such as, for example, synergetic outputs from undifferentiated inputs and patterns of regularity without specific regulators. (Schmidt, Treffner & Turvey 1991, 120).

Manche Unkorrektheiten können so entdeckt werden, die andere Modelle unbemerkt in sich tragen. Hervorstechendstes Merkmal von VITE ist, daß annähernd natürliche Bewegungsmuster durch Interaktion innerhalb des Netzes entstehen.



Demgegenüber stehen Beschränkungen, die die Attraktivität des Modells mindern. Zunächst ist dies die Einschränkung auf Greifbewegungen. Viele sportliche Bewegungen, vor allem Schlagbewegungen, bei denen es darauf ankommt, zu einem bestimmten Zeitpunkt eine möglichst hohe Geschwindigkeit zu haben, können nicht modelliert werden. Auch bei zyklischen Bewegungen oder Bewegungsfolgen erscheint dies aufgrund der Eigenschaften des *GO*-Signals fragwürdig. Keine Aussagen können im Modell über den Lernverlauf gemacht werden. Die biologische Plausibilität beschränkt sich auf das Prinzipielle, daß von verteilter Repräsentation im Gehirn ausgegangen wird.

Bei aller Kritik an den Beschränkungen kann jedoch gesagt werden, daß das VITE-Modell für das Verstehen der motorischen Kontrolle von Bewegungen ein Schritt nach vorne ist. Fast alle alternativen Modelle (*mass-spring*, *minimum-jerk*, synergetische Modelle), die sich mit der Kontrolle von Greifbewegungen beschäftigen, teilen ähnliche Beschränkungen. Den Vorsprung von VITE erklärt Rosenbaum (1991, 382):

But where synergeticists *describe* emergent properties in already-existing natural systems, network modelers *create* systems whose emergent properties are meant to be adaptive for the tasks at hand.

und deswegen

Network modeling provides a way (now proven) [i.e. with Bullock and Grossberg's VITE-Model] of exploring how those emergent properties might arise (Rosenbaum 1991, 368).

### 3.2 Jordans Netz

Das zweite in der bewegungswissenschaftlichen Literatur (z.B. Rosenbaum 1991, 320-321; Morasso & Sanguineti 1992, 5-9; Marteniuk 1992, 112-114; Keele & Jennings 1992, 127-128) besprochene Netzwerkmodell stammt von Michael Jordan, einem amerikanischen Forscher am Massachusetts Institute of Technology.

### 3.2.1 Architektur

Jordans Ziel ist es, ein Netzwerk zu entwickeln, das für ein auf höheren Verarbeitungsebenen erzeugtes erwünschtes Bewegungsergebnis die entsprechende Bewegung, d.h. die entsprechenden Steuerungssignale an die Muskulatur findet.<sup>5</sup> Dabei ergibt sich das Problem, daß das tatsächliche Bewegungsergebnis immer eine Interaktion zwischen Zustand der Umwelt und Bewegung des Lernenden ist (s. Abb. 10).

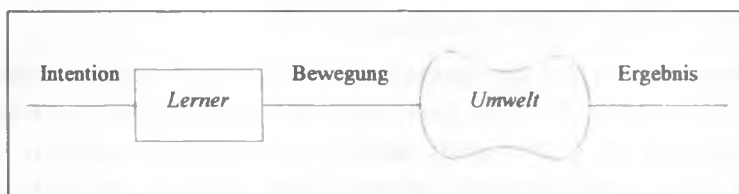


Abb. 10: Beziehung zwischen Lerner und Umwelt (nach Jordan & Rumelhart 1992, 315).

Der Lerner möchte ein bestimmtes Bewegungsziel  $z^*$  erreichen. Dazu produziert er ein Steuerungssignal  $x$  an die Muskulatur (gleichgesetzt mit der daraus entstehenden Bewegung). Diese Bewegung hat in Abhängigkeit von dem Zustand der Umgebung  $s$  ein Bewegungsergebnis  $z$  zur Folge.

Konsequenterweise definiert nun Jordan das Bewegungsergebnis  $z$  als die Funktion von Umgebungszustand und Bewegung:

$$z = f(s, x), \quad (9)$$

die produzierte Bewegung als die Ausgabe eines Netzes, die abhängig von einem Bewegungsplan  $p$ , einem internen Zustand  $q$  und den Gewichten  $w$  innerhalb des Netzes ist:

<sup>5</sup> Jordan unterscheidet hier nicht zwischen Bewegung („action“, Jordan & Rumelhart 1992, 308) und Bewegungskommandos („motor program output“, Jordan 1989, 799). Dies ist aber in seinem Modell auch nicht wichtig, da er prinzipiell von einer *many-to-one*-Abbildung von Bewegung (oder Bewegungskommandos) zum Bewegungsergebnis ausgeht. Somit würden also durch eine mögliche *many-to-one*-Abbildung zwischen Bewegungskommandos und Bewegungen zusätzlich entstehende Freiheitsgrade keine prinzipiellen Auswirkungen auf Jordans Modell haben.

$$\mathbf{x} = g(\mathbf{q}, \mathbf{p}, \mathbf{w}). \quad (10)$$

In Anlehnung an Abbildung 10 wird  $f$  als die Lernerfunktion,  $g$  als die Umweltfunktion bezeichnet. Für den Fall, daß die Lernaufgabe perfekt gelöst ist, wird also das gewünschte Bewegungsergebnis  $\mathbf{z}^*$  immer erreicht werden, also gilt immer  $\mathbf{z}^* = \mathbf{z}$ . Das bedeutet, daß  $g(\mathbf{z}^*) = \mathbf{x}$ ,  $f(\mathbf{x}) = \mathbf{z}$ , bzw. wegen der Gleichheit von  $\mathbf{z}$  und  $\mathbf{z}^*$   $g(\mathbf{z}) = \mathbf{x}$ ,  $f(\mathbf{x}) = \mathbf{z}$  ist. Der Lernende transformiert also aus  $\mathbf{z}^*$  ( $= \mathbf{z}$ ) die Bewegung  $\mathbf{x}$ , die in der Umwelt das Ergebnis  $\mathbf{z}$  ( $= \mathbf{z}^*$ ) bewirkt. Daraus folgert Jordan, daß es beim Erlernen von Bewegungen darum geht, die *Inverse*, das heißt die Umkehrrelation der Umweltfunktion zu lernen (vgl. Jordan & Rosenbaum 1989, 748ff, Jordan & Rumelhart 1992, 314ff). Dieses Netz nennt Jordan *inverse model*. Es soll hier im folgenden, da seine Ausgabe die Steuersignale an die Muskulatur modelliert, *Bewegungssteuerungsnetz* genannt werden.

Das Problem hierbei ist, daß Fehler nur auf der Ebene des Bewegungsergebnisses, nicht jedoch auf der Ebene der Bewegungen verfügbar sind. Somit ist das direkte Modellieren der Inversen nur durch unüberwachtes Lernen, bei dem keine Rückmeldung eines Fehlers benötigt wird, möglich. Unüberwachtes, also selbstorganisierendes Lernen, wie beispielsweise bei AVITE in Kapitel 3.1.3 vorgestellt, erlaubt jedoch keine Einschränkung des Ausgabebereichs, kein zielgerichtetes Lernen, was Jordan bei der Modellierung motorischen Lernens aber wünschenswert erscheint. Ein zusätzlicher Nachteil ist, daß selbstorganisierende Netzwerke keine non-konvexen Probleme lösen können. Dies sind Probleme, bei denen Vektoren innerhalb des „Aufmerksamkeitskegels“ (vgl. S. 48) nicht zur Lösungsmenge gehören. Jordan und Rumelhart (1992, 328) weisen nach, daß es schon bei einfachen Greifbewegungen non-konvexe Probleme gibt.

Um überwachtes Lernen zuzulassen, behilft sich Jordan mit einem Kunstgriff: Er entwickelt ein *forward model*, ein Modell der Umweltfunktion, das lernt, für gegebene Steuersignale die Bewegungsergebnisse vorherzusagen. Seine Eingaben sind der momentane Umgebungszustand und die Steuersignale an die Muskulatur, seine Ausgabe das

Bewegungsergebnis. Es dient also dazu, die Konsequenzen von Bewegungssteuerungssignalen vorherzusehen, es modelliert die Bewegungsvorstellung. Im folgenden wird dieses Netz *Bewegungsvorstellungsnetz* genannt.

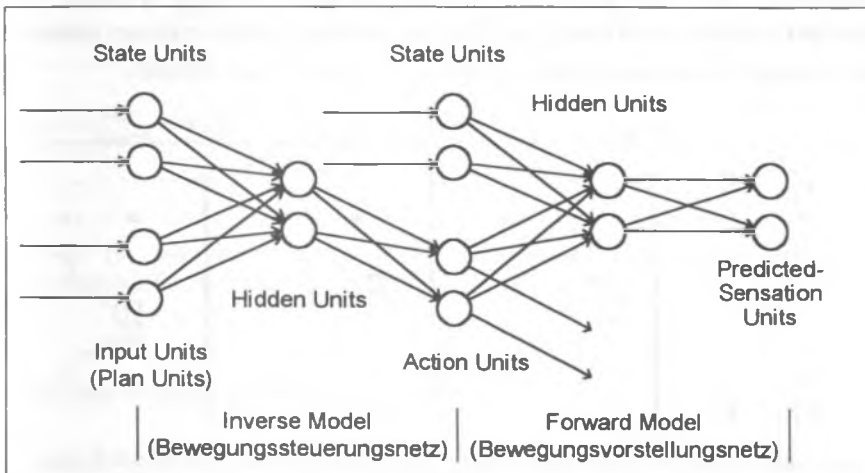
Das Netz modelliert das Lernen der Bewegungsvorstellung mit einem modifizierten *backpropagation*-Algorithmus, indem eine einzelne Bewegung als Eingabe dient, das vorhergesagte Bewegungsergebnis als Ausgabe und der Vorhersagefehler (*prediction error*, die Differenz zwischen vorhergesagtem und tatsächlichem Bewegungsergebnis) zurückpropagiert wird. In Erweiterung des Standard-*backpropagation* multipliziert Jordan den Fehler vor der Rückführung mit zwei positiven finiten Matrizen. Eine ermöglicht durch ihre Manipulation die Einführung von *don't care*-Bedingungen, das sind Bedingungen, bei denen der von einem bestimmten Neuron zurückgemeldete Fehler nicht berücksichtigt wird. Mit der zweiten können die Fehlerkomponenten einzeln gewichtet werden (vgl. Jordan 1992, 400ff) Diese Veränderungen werden bei der Modellierung zusätzlicher Randbedingungen, die das Problem der Freiheitsgrade lösen helfen, wichtig.

Nachdem die Bewegungsvorstellung im Netz gelernt worden ist, kann das Lernen der Steuersignale beginnen. Dazu wird das Bewegungssteuerungsnetz vor das Bewegungsvorstellungsnetz plziert. Die Ausgabeneuronen des Bewegungssteuerungsnetzes sind zugleich Eingabeneuronen in das Bewegungsvorstellungsnetz (s. Abb. 11). Der Ausführungsfehler (*performance error*, die Differenz zwischen gewünschtem und tatsächlichem Bewegungsergebnis) wird zunächst durch das Bewegungsvorstellungsnetz zurückpropagiert, wobei die Gewichte konstant gehalten werden. Das Ergebnis dieser Fehlerrückführung liefert für jede Komponente des Bewegungsvektors ihren Anteil am Ausführungsfehler.<sup>6</sup> Dieser Anteil kann weiter zurückgeführt werden, wobei im Bewe-

---

<sup>6</sup> Es handelt sich eigentlich um den durch das Bewegungsvorstellungsnetz „geschätzten“ Anteil am Ausführungsfehler. Nur wenn es fehlerfrei ist, entspricht der geschätzte Anteil auch dem wirklichen. Bei einer fehlerhaften (aber nicht völlig falschen) „Bewegungsvorstellung“ gelingt jedoch das Lernen trotzdem, wenngleich es länger dauert. Geometrisch interpretiert trägt das Bewegungsvorstellungsnetz zur Errechnung des steilsten Gradienten in der Fehlerlandschaft bei. Wenn dieser nur

gungssteuerungsnetz die Gewichte entsprechend verändert und so die Steuersignale gelernt werden. Dabei spielt es keine Rolle, daß es viele verschiedene Lösungsmöglichkeiten gibt, daß viele Ausgaben des Bewegungssteuerungsnetzes zu dem richtigen Ergebnis führen, daß also viele Bewegungen das gewünschte Ergebnis zur Folge haben. Benutzt man die geometrische Interpretation des Problems (s. S. 47), so gibt es in der Fehlerlandschaft viele globale Minima mit dem Fehler Null, und in eins davon „rollt die Kugel“. Natürlich werden nicht alle Lösungen von außen betrachtet gleich sinnvoll sein. Zusätzliche Randbedingungen, die sinnvolle von sinnlosen Lösungen trennen (z.B. bei einem Torschuß im Fußball „so hart wie möglich“), müssen gefunden werden.

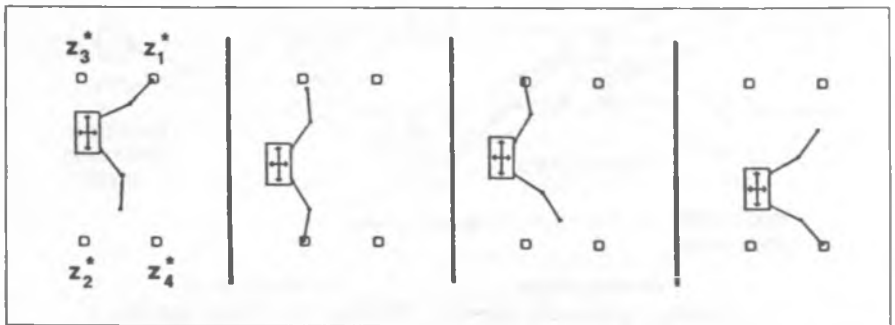


**Abb. 11:** Ein vorwärtsgerichtetes Netzwerk aus Inversenmodell und Vorwärtsmodell (aus Jordan & Rumelhart 1992, 321). Die *action units* sind die Ausgabestellen.

Dieses Grundmodell kann nur einzelne diskrete Bewegungen einer Bewegungsklasse modellieren. Üblicherweise und vor allem im Sport wird motorisches Handeln jedoch durch Bewegungsfolgen charakterisiert. In einer einfachen Simulation sequentieller Bewegungsfolgen

modelliert Jordan eine Hand mit zwei Fingern, die nacheinander vier Zielpunkte ansteuern soll. Das Ergebnis der Simulation zeigt Abbildung 12. Dazu muß das Netz viermal durchlaufen werden, wobei bei jedem Durchlauf ein anderes Zielfeld in Form eines „Plan“-Vektors eingegeben wird.

Zusätzlich zu diesem „Plan“-Vektor muß auch der aktuelle Zustand der Hand und für den Fall, daß sich die Umwelt ändert, auch der Zustand der Umwelt bekannt sein. Dies wird erreicht, indem Verbindungen von der Ausgabeschicht des Bewegungssteuerungs- und des Bewegungsvorstellungnetzes zu zusätzlichen Statusvektoren in der Eingabeschicht des Bewegungssteuerungsnetzes implementiert werden (s. Abbildung 13). Soll nun die Bewegungssequenz gelernt werden, muß das Netz viermal rückwärts durchlaufen werden, dabei müssen jedesmal die Gewichte im Bewegungssteuerungsnetz korrigiert werden.



**Abb. 12:** Ergebnis einer Netzsimulation einer „Hand“ mit zwei „Fingern“ (aus Jordan 1989, 815). Nacheinander sind z1 bis z4 anzusteuern.

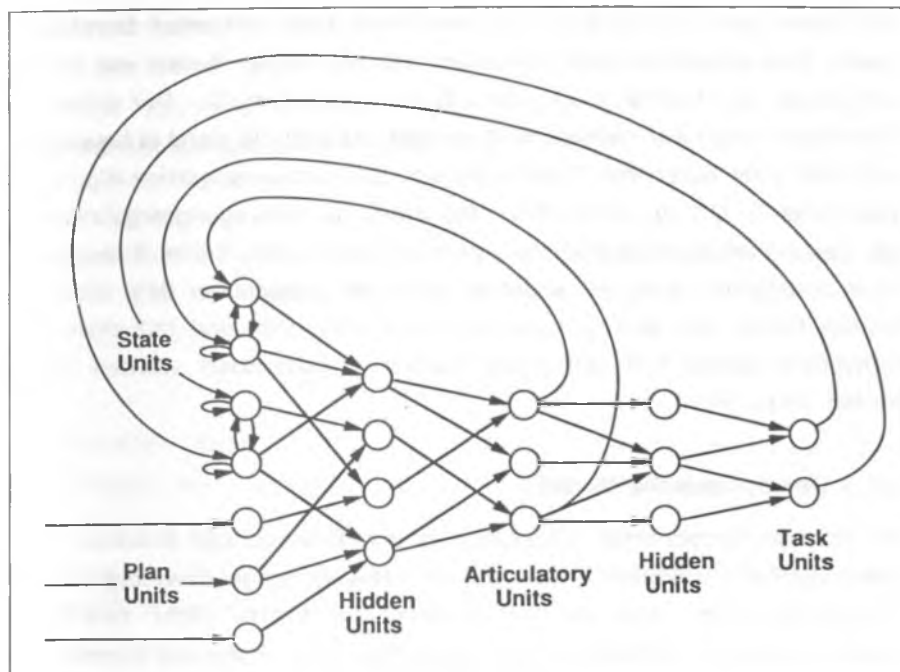


Abb. 13: Rekurrentes Netzwerk zur Modellierung von Bewegungssequenzen (aus Jordan 1989, 808).

### 3.2.2 Das Problem der Freiheitsgrade

Bei der Simulation von Bewegungsfolgen kann das Problem der Freiheitsgrade sinnvoll gelöst werden. Um im Netzwerk Einfluß auf die Auswahl einer Lösung aus der Menge der möglichen Lösungen zu bekommen, bedarf es der Einführung zusätzlicher Randbedingungen. Wie in Abbildung 12 zu sehen, findet das Netz unter den unendlich vielen verschiedenen Möglichkeiten eine anscheinend „natürliche“. Dies erreicht Jordan durch Hinzufügen eines *smoothness constraint*, einer Randbedingung, die fließende, geschmeidige Bewegungen garantieren soll. Diese Geschmeidigkeitsbedingung addiert auf der Ebene der Ausgangsneuronen des Bewegungssteuerungsnetzes zu dem Ausführungsfehler einen weiteren Fehlerterm, der sich aus der Differenz zwischen

dem Wert beim vorherigen und dem Wert beim aktuellen Durchlauf ergibt. Dies entspricht dem Kriterium des minimalen Rucks von Flash und Hogan (s. Kapitel 1.3.1) für diskrete Zeitintervalle. Der addierte Fehlerterm wird zwar niemals Null werden, da sich die Hand ja bewegen soll, wird aber durch die Funktionsweise des *backpropagation*-Algorithmus minimal. Um zu verhindern, daß statt des Bewegungsergebnisses die Geschmeidigkeitsbedingung optimiert wird, also keine Bewegung mehr stattfindet, muß der addierte Fehlerterm gegenüber dem rückgeführten Fehler des Bewegungsergebnisses klein sein und mit einer im Lernverlauf gegen Null gehenden Variablen multipliziert werden (vgl. Jordan 1992, 407).

### 3.2.3 Das dynamische Modell

Die bis hier betrachteten Simulationen von Bewegungen beruhen auf *kinematischen* Modellen, bei denen Bewegungsgeschwindigkeiten, -beschleunigungen und die sie auslösenden Kräfte nicht beachtet werden. In einem weiteren Schritt versuchen nun Jordan und Rumelhart (1992), Bewegungen *dynamisch* an dem typischen Beispiel einer Armbewegung zu modellieren. Dazu ersetzen sie das kinematische Bewegungsvorstellungsnetz der Umwelt durch ein dynamisches *forward model* des Arms, das (kinematische) Bewegungssteuerungsnetz wird folgerichtig zum (dynamischen) *inverse model* des Arms, oder, mit Jordans Worten, zum *controller* (vgl. Abb. 14).

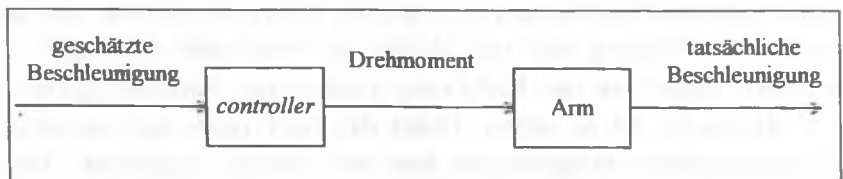


Abb. 14: Die Funktion des Bewegungssteuerungs- und des Bewegungsvorstellungsnetzes übertragen auf die Dynamik eines Arms (nach Jordan & Rumelhart 1992, 326).



Die Lernaufgabe des Bewegungsvorstellungsnetzes des Arms besteht darin, aus momentaner Position  $\mathbf{q}$  und Winkelgeschwindigkeit  $\dot{\mathbf{q}}$  sowie dem auf den Arm einwirkenden Drehmoment  $\tau$  die sich ergebende Winkelbeschleunigung  $\ddot{\mathbf{q}}$  zu schätzen. Dafür muß in der Gleichung

$$\hat{\ddot{\mathbf{q}}} = f(\mathbf{q}, \dot{\mathbf{q}}, \tau, \mathbf{w}) \quad (11)$$

der Gewichtevektor  $\mathbf{w}$  so gewählt werden, daß die Funktion so gut wie möglich  $\ddot{\mathbf{q}}$  in der nach  $\ddot{\mathbf{q}}$  aufgelösten Standarddifferentialgleichung für dynamische Systeme

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \tau \quad (12)$$

approximiert (dabei ist  $\hat{\ddot{\mathbf{q}}}$  die geschätzte Winkelbeschleunigung,  $\mathbf{M}(\mathbf{q})$  eine Matrix mit Trägheitsmomenten,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  eine Matrix mit Coriolis- und Zentripetalkräften,  $\mathbf{G}(\mathbf{q})$  ein durch Schwerkraft bewirkter Drehmomentvektor und  $\tau$  das resultierende Drehmoment; beide Gleichungen aus Jordan & Rumelhart 1992, 333f).

Für die Anwendung des *backpropagation*-Algorithmus' ergeben sich dabei zwei Probleme:

1. Drehmomente und Kräfte, die auf einen Körper einwirken, sind stetig über der Zeit, sie verändern sich nicht abrupt, sondern kontinuierlich. Das Lernen durch *backpropagation* funktioniert aber besser, wenn die Eingabewerte zufällig streuen.
2. Das Neuronenmodell, für den *backpropagation* entwickelt wurde, ist zeitdiskret.

Das zweite Problem kann relativ leicht gelöst werden, indem die Differentialgleichung, die die Dynamik des Arms beschreibt, durch bekannte Algorithmen in eine zeitdiskrete Näherungsfunktion überführt wird. Zur Lösung des ersten Problems erweitern Jordan und Rumelhart das Modell um einen *feedback-controller*. Die einzusetzenden Kräfte werden gemeinsam von dem *feedforward*- und dem *feedback-controller* ermittelt. Zu Anfang des Lernens erzeugt der *feedforward-controller* kleine, zufällige Drehmomente, und die Hauptlast der Bewegungssteuerung liegt auf dem *feedback-controller*, aber mit zunehmender Verbesserung der

Steuerung des Arms nimmt die „Vorwärtskontrolle“ zu und damit gleichzeitig die „Rückmeldungs-Kontrolle“ ab, da die Fehler, die zurückgemeldet werden, wegen der Verbesserung des Bewegungssteuerungsnetzes von vornherein kleiner sind.

Zu einem dynamischen Modell der Bewegung gehört auch ein dynamisches Modell der Umgebung. Dazu schreiben Jordan und Rumelhart (1992, 343):

In many problems involving delayed temporal consequences, however, it is neither feasible nor desirable to learn a dynamic forward model of the environment, either because the environment is too complex or because solving the tasks at hand does not require knowledge of the evolution of all the state variables.

Hier werden die Leistungsgrenzen künstlicher neuronaler Netzwerke deutlich. Während die Simulation der dynamischen Bewegungskontrolle eines eingelenkigen Arms noch gelingt, ist die Dynamik der Umgebung in der Regel so komplex, daß nicht einmal ein vereinfachtes Modell auf den bisherigen Rechnern simuliert werden kann.

### 3.2.4 Einordnung in den sportwissenschaftlichen Zusammenhang

Ausgangspunkt der Überlegungen von Jordan und Kollegen ist die Theorie generalisierter motorischer Programme (GMP-Theorie) und Schmidts Schematheorie über das Lernen diskreter motorischer Fertigkeiten.

In einer frühen Veröffentlichung bezieht sich Jordan (1989, 797) explizit auf die GMP-Theorie:

This chapter describes ongoing work on a theory of action that combines elements of both the generalized motor program perspective and the dynamical systems perspective. (...) The approach that is developed here has similar goals to those of generalized motor program theory but is more suited to problems involving excess degrees of freedom and goal-directed learning.

Das *inverse model* – also das Bewegungssteuerungsnetz – heißt dort auch *motor program subnetwork* (a. a. O., 807). Es entspricht allerdings nicht nur der Funktionsweise eines GMPs, sondern besitzt auch Eigenschaften von Schmidts Erinnerungsschema: Der regelhaften Beziehung

zwischen Ausgangsbedingungen, gewünschtem Bewegungsergebnis und den Handlungsspezifikationen bei Schmidt entspricht die Beziehung zwischen Statusvektor, Planvektor und der Ausgabe der *action units* – also den Steuerungssignalen an die Muskulatur – in Jordans Netz. Eine Ähnlichkeit zwischen dem Wiedererkennungsschema und dem Bewegungsvorstellungnetz läßt sich ebenfalls erkennen: Schmidts Wiedererkennungsschema ist der regelhafte Zusammenhang von erwarteten sensorischen Konsequenzen einerseits und Ausführungsbedingungen und gewünschtem Bewegungsergebnis andererseits; in Jordans Bewegungsvorstellungnetz entstehen die erwarteten sensorischen Konsequenzen aus dem Statusvektor und dem Output der *action units*.

Eine Korrektur fehlerhaft gebildeter Schemata erfolgt bei Schmidt aus dem Vergleich zwischen erwarteten und tatsächlichen sensorischen Konsequenzen, bei Jordan wird das Bewegungsvorstellungnetz durch eben diesen Vergleich gelernt, das Bewegungssteuerungsnetz jedoch durch den Fehleranteil der einzelnen Steuersignale an dem beobachteten Bewegungsfehler. Damit löst Jordan ein Problem, das in der Schematheorie nur oberflächlich behandelt wird: Wie kommt man von der Kenntnis des Resultats zu den Handlungsspezifikationen? Hierzu bemerken Jordan und Rosenbaum (1989, 754):

The issue has perhaps been overlooked because both of these theories [the schema theory and the closed-loop theory of Adams (1971)] were designed to address data from discrete, single-degree-of-freedom movements. In such movements the problem of converting from knowledge of results to a change in the response specification is not particularly difficult. In tasks with many degrees of freedom, however, this conversion is arguably the most difficult part of the learning process.

Die Schematheorie und Jordans Modell unterscheiden sich aber auch dadurch, daß Schmidts Theorie lineare Zusammenhänge zwischen der Wahl der Parameter und dem Ergebnis der Bewegung postuliert, während Jordans Netz nichtlineare Zusammenhänge erfaßt. Dies hat erhebliche Konsequenzen. So ist ein GMP gerade durch diesen linearen Zusammenhang wohldefiniert. Alle Bewegungen einer Bewegungsklasse, innerhalb derer die zeitliche Abfolge der Muskelimpulse und die relativen Krafteinsätze linear sind, werden von einem GMP gesteuert.

Auf die dadurch entstehenden Probleme, z.B. das Freiheitsgrade-Speicherplatz-Dilemma, wurde schon in Kapitel 1.4 hingewiesen. Sie können auch nicht dadurch behoben werden, daß man annimmt, Schmidt habe zwar seine Theorie am Beispiel linearer Zusammenhänge veranschaulicht, sie sei aber auch auf nichtlineare Zusammenhänge übertragbar. Genau dies ist sie aber nicht!

Bei nichtlinearen Schemaregeln könnten beispielsweise alle Fingerbewegungen als ein Programm definiert werden, deren unterschiedlicher Einsatz durch Parameter gesteuert würde. Genauso gut ließen sich die Arm- oder Oberkörperbewegungen als ein GMP definieren. Im Extremfall könnte ein einziges, sehr komplexes Programm mit nichtlinearen Parametern das gesamte Bewegungsverhalten des Menschen abbilden. Durch ein Zulassen nichtlinearer Schemaregeln würden die Grenzen zwischen Programm und Parameter also völlig beliebig. Dadurch verlöre Schmidts Schematheorie ihre Vorhersagekraft und ihre begriffliche Klarheit, also gerade die Eigenschaften, die sie so attraktiv machen.

Jordan verzichtet auf die Unterscheidung zwischen Programm und Parameter. Der Geltungsbereich des Bewegungssteuerungsnetzes bleibt undefiniert. Auch hier wäre es denkbar, die Gesamtheit aller menschlichen Bewegungen als *eine* (höchst komplizierte) nichtlineare Differentialgleichung auszudrücken, die von einem Netz mit genügend Neuronen in der verdeckten Schicht gelernt oder approximiert werden könnte. Die Anzahl der Neuronen ginge allerdings vermutlich in die Millionen und die Rechenzeit zum Erlernen dieser Gleichung betrüge Jahrzehnte, selbst wenn wesentlich effizientere Lernalgorithmen vorausgesetzt werden. Dies täte aber dem Modellcharakter keinen Abbruch.

Diese Möglichkeit, nichtlineare Zusammenhänge zu modellieren, ermöglicht genau den schon oben zitierten Brückenschlag zwischen dem *motor approach* und dem *action approach*. So definieren Schöner, Zanone und Kelso „learning as change of coordination dynamics“ (1992, 29) und Lames fragt „Die komplexe sportliche Leistung – ein nichtlineares dynamisches System?“ (1991, 1). Jordans Modell stellt also ein Instrument dar, mit dem die abstrakte Modellierung des Bewegungsverhaltens durch nichtlineare dynamische Differentialgleichungen

in einer konkreten Simulation greifbar gemacht werden.

### 3.3 Vergleich von Jordans Modell und VITE

Bullock und Grossbergs Netzwerkmodell und Jordans rekurrentes Netz teilen viele Gemeinsamkeiten. Beide Implementationen benutzen abstrakte Modelle von Neuronen, die ohne globale Information allein durch die Dynamik ihrer Interaktion das Entstehen von Steuerungssignalen an die Muskulatur simulieren. Dabei wird die Bewegung durch zwei auf höherer Verarbeitungsebene erzeugten Vektoren beeinflusst: Intentionale Komponenten der Bewegungssteuerung werden bei Bullock und Grossberg durch das *GO*-Signal modelliert, bei Jordan durch den Plan-Vektor. Der Einfluß von durch die Umgebung festgelegte Randbedingungen auf die Bewegung werden bei VITE durch das Zielpositionskommando *TPC*, bei Jordan durch den Status-Vektor nachgebildet. Rückmeldungen können bei beiden Netzwerken verwendet werden, jedoch konzentrieren sie sich auf das schwierigere Problem der Bewegungskontrolle ohne Steuerung durch Rückmeldungen. Dabei wird statt ihrer eine vor Bewegungsbeginn verfügbare Schätzung des Bewegungsverlaufs auf der Grundlage der efferenten Signale (Differenzvektor *DV* bzw. Bewegungsvorstellungnetz) verwendet. Diese Schätzungen müssen in einer Phase des „motorischen Stammelns“ (Gaudiano & Grossberg 1992, 142; Jordan 1989, 799) zumindest teilweise durch den Vergleich mit reafferenten Signalen gelernt werden, bevor vorwärtskontrollierte Bewegungen ausgeführt werden können. Bullock und Grossberg beschränken sich bei der Umgebungsinformation auf die Zielposition des zu greifenden Gegenstands, bei Jordan ist die Menge der einflußnehmenden Umgebungsvariablen nicht festgelegt.

Der Hauptunterschied zwischen beiden Netzen liegt beim Lernen. Bei Gaudiano und Grossbergs AVITE wird die Zuordnung von Änderungen der Muskelaktivierungen zu den Änderungen der Positionen der Gliedmaßen im Raum gelernt. Sie benutzen ein selbstorganisierendes Lernverfahren, das Netz entdeckt damit die Gemeinsamkeiten zwischen

Muskelaktivierungen und Positionen der Gliedmaßen. Dieses Verfahren findet seine Grenzen zum einen darin, daß verschiedene Positionen der Gliedmaßen durch die unterschiedlichsten Muskelaktivierungen erreicht werden können (das Problem der inversen Dynamik, vgl. Kapitel 1.3.1), zum anderen, daß unter Umständen benachbarten Positionen keine ähnlichen Muskelaktivierungen zugrunde liegen. Diese Probleme löst Jordan in seinem Modell dadurch, daß er überwachtes Lernen zuläßt. Dabei können zusätzliche Randbedingungen eingeführt werden, die das Problem der inversen Dynamik lösen. Durch das Überwachen des Lernens entfällt auch bei ausreichendem Üben das Problem, daß falsche Interpolationen des Netzes nicht aufgedeckt werden.

### 3.4 Andere Modelle

In diesem Kapitel werden kurz zwei andere Netzwerke beschrieben, um die vielfältigen Möglichkeiten der Modellierung deutlich zu machen.

#### 3.4.1 Das M-Net

Auf der Basis des *equilibrium-point model* (Gleichgewichtspunkt-Modell), einer Erweiterung des in Kapitel 1.3.2 kurz vorgestellten Masse-Feder-Modells, entwickelten Morasso und seine Mitarbeiter (Morasso 1990, 1992; Morasso & Sanguineti 1992) ihr sogenanntes *M-net*<sup>7</sup>. Zwei neue Ideen fließen in ihre Netzarchitektur ein, die Modellierung von passiven Bewegungen und die Organisation des Netzes in Modelle von Muskeln, Bändern und Knochen.

Passive Bewegungen sind aus Sicht der Vorwärtskontrolle recht problematisch. Sie werden nicht durch aktiven Muskeleinsatz, sondern durch äußere Kräfte wie Gravitations- oder Fliehkräfte erzeugt. Damit entziehen sie sich der *feedforward*-Kontrolle, bei der die Position der

<sup>7</sup> Dieser Name wird als Abkürzung für *motor relaxation network* eingeführt, ein so wenig sprechender Name, daß man vermuten möchte, die Intention des Verfassers sei, die Abkürzung doch lieber mit „Morasso-net“ zu assoziieren.

kontrollierten Körperteile auf der Grundlage der Muskeleinsätze geschätzt wird. Morasso sieht eine natürliche Lösung dieses Problems darin, daß ein zentraler neuronaler Mechanismus diese passiven Bewegungen nachvollzieht (Morasso 1992, 170). Das *M-net* simuliert also virtuelle Kraftimpulse, die die Muskeln liefern müßten, um passive Bewegungen aktiv durchzuführen. Diese Simulation der passiven Bewegung „is basically a *degree-of-freedom cruncher*“ (a. a. O., 171), da es bei einer passiven Bewegung keine Freiheitsgrade gibt, sondern sie immer durch äußere und innere Kräfte vorgegeben wird.

Das *M-net* besteht aus drei verschiedenen Einheiten. *S-units*, *M-units* und *L-units*. Mit den *S-units* werden Skelettsegmente (oder Knochen) modelliert. Sie werden als starre Körper angesehen, an denen interne und externe Kräfte angreifen. Im Modell werden interne Kräfte von Muskeln, Bändern und der Schwerkraft erzeugt, durch externe Kräfte soll die beabsichtigte Bewegungsrichtung ausgedrückt werden. Die externen Kräfte werden von einem *motor planner* außerhalb erzeugt und sind die globale Eingabe des Netzes.

Muskeln werden durch die *M-units* modelliert. Sie greifen an zwei durch ein Gelenk verbundenen Knochen an. Als Eingabe dient der Angriffspunkt am Knochen, als Ausgabe produzieren sie Kraftvektorraupere, die sich zu Null addieren. Je ein Kraftvektor greift an je einem mit dem Muskel verbundenen Knochen an. Das Verhalten der *M-units* kann als Funktion von Muskellänge und -anspannung beschrieben werden.

Die *L-units* modellieren die Gelenke mit ihren biomechanischen Begrenzungen. Sie können als Federn angesehen werden, die bis zu einer bestimmten Länge dehnbar sind. Je nach Gelenkart müssen ein oder mehrere Federn spezifiziert werden, um ein Gelenk adäquat zu modellieren.

Diese verschiedenen *units* werden nun nach dem Prinzip eines Hopfield-Netzwerks (vgl. S. 41) zusammengeschaltet, wobei die *S-units* die Rolle der Neuronen im Hopfield-Modell übernehmen, die *L-* und *M-units* die der Gewichte, die im Gegensatz zum Hopfield-Modell jedoch nichtlinear und veränderbar sind. Der Zustand eines Hopfield-Netzwerks kann durch

eine Energiefunktion beschrieben werden. Sie ist die Summe der Produkte aus allen Erregungen mit den dazugehörigen Gewichten. Erreicht diese Energiefunktion ein (lokales) Minimum, ist der Zustand des Netzes stabil. Der Übergang von einem instabilen zu einem stabilen Zustand wird *relaxation*, zu deutsch „Entspannung“ genannt. Die Besonderheit des *M-net* ist nun, daß in einer „aktiven Phase“ die *M-units* und damit die Gewichte verändert werden, in einer „passiven Phase“ dann das Netz sich zu dem neuen Minimum hin „entspannt“. Dieses ständige Hinterherhinken des Netzzustandes einer sich immer wieder verändernden Energielandschaft repräsentiert das Fortschreiten einer Bewegung auf einer Bewegungsbahn.

Wie das VITE-Modell von Bullock und Grossberg kann auch das *M-net* weder Ziel noch Geschwindigkeit der Bewegung festlegen. Auch hier wird davon ausgegangen, daß dies auf höherer Verarbeitungsebene geschehen muß. Sind aber Ziel und Geschwindigkeit vorgegeben, findet das *M-net* die dazugehörige Bewegungsbahn.

### 3.4.2 Das Kaskaden-Netz

Ein dem Netz von Jordan ähnliches Modell entwickelte in Osaka der Japaner Mitsuo Kawato mit seinen Kollegen (Kawato, Furukawa & Suzuki 1987; Kawato 1990a und b; Kawato, Maeda, Uno & Suzuki 1990; Hirayama, Kawato & Jordan 1993). Im Unterschied zu Jordans Netz gibt es bei Kawato keine rekurrenten Verbindungen, sondern hintereinandergeschaltete Netze. In einer ersten Lernphase wird auch hier eine Bewegungsvorstellung gelernt, die den momentanen Bewegungszustand in Abhängigkeit von momentanem Ort, aktueller Geschwindigkeit und einwirkender Drehmomente lernt. Um dynamische Modelle zu erlernen, werden die die Bewegung beschreibenden Differentialgleichungen in eine zeitdiskrete Form gebracht. Als Eingabe in das erste Netz dienen die Orts-, Geschwindigkeits- und Kraftvektoren zum Zeitpunkt Null. Die Ausgabe des Netzes sind Orts- und Geschwindigkeitsvektoren zum Zeitpunkt  $\Delta t$ . Sie wird mit den Vektoren der tatsächlichen Bewegung des kontrollierten Objekts verglichen und mit dem



*backpropagation*-Algorithmus gelernt. Die Ausgabevektoren des ersten Netzes werden in ein zweites Netz eingespeist, dazu die Kraftvektoren zum Zeitpunkt  $\Delta t$ , die von außen geliefert werden. Dieses zweite Netz produziert dann Schätzungen der Orts- und Geschwindigkeitsvektoren zum Zeitpunkt  $2\Delta t$ , die dann genauso gelernt werden. Dieses Verfahren wird nun für alle Zeitpunkte kaskadenförmig weitergeführt, für jeden Zeitpunkt existiert also ein Netz. Die einzelnen Netze bestehen aus vier Schichten. Die ersten drei Schichten bilden ein bekanntes mehrschichtiges Netz, das die Geschwindigkeits- und Ortsänderungen lernt, in der vierten Schicht werden einfach die Eingabevektoren zu den Ausgabevektoren addiert, wobei die neuen Geschwindigkeits- und Ortskoordinaten entstehen.

Nachdem in der Lernphase ein Vorwärtsmodell gelernt wurde, werden in der *pattern-generating phase* die Drehmomentvektoren miteinander gekoppelt. Nach Art des oben erwähnten Hopfield-Netzes entspannt sich das Netz nun zu einem energetisch stabilen Zustand. Da die Energie hier über die Drehmomentvektoren definiert wird, entspricht dieses Verfahren dem *minimum torque change*-Kriterium aus Kapitel 1.3.1.

## 4 Simulation

Ziel dieses Abschnitts soll es sein, ein Beispiel für eine abstrakte Modellierung eines möglichst sporttypischen Bewegungsmusters mit einem neuronalen Netz zu geben und dabei zu überprüfen,

- ob die für das menschliche Bewegungslernen empirisch gefundenen Gesetzmäßigkeiten auch für ein konnektionistisches Modell gelten
- und ob das in Kapitel 1.1 schon ausführlich vorgestellte Problem der Freiheitsgrade, als das wichtigste theoretische Problem der motorischen Kontrolle in einem solchen Modell, sinnvoll angegangen werden kann.

Sollte dies zutreffen, so sind die Konsequenzen aus einem solchen Ergebnis allerdings umstritten. Anhänger des Konnektionismus' schließen daraus, daß dann neuronale Netze als sinnvolle Modelle für bewegungssteuernde Prozesse anzusehen sind und aufgrund der bekannten Mechanismen in den Netzwerken die empirischen Befunde theoretisch neu erklärt werden können. Seine Gegner sind der Ansicht, daß durch eine solche Modellbildung auf neuronaler Ebene keine Konsequenzen für die Erklärung psychischer Kategorien erwartet werden können. Diese Diskussion soll aber erst in Kapitel 5 aufgegriffen werden.

### 4.1 Grundsätze

Bei der Simulation von menschlichem Verhalten auf neuronalen Netzwerken werden empirisch oder theoretisch gewonnene Verhaltensdaten durch das Netzwerk nachgeahmt. Ein künstliches neuronales Netzwerk besteht aus vielen verschiedenen abstrakten Neuronen, die als simple Prozessoren aufgefaßt werden können, und ihren Verbindungen untereinander. Dies sind die „Hardware“-Komponenten des Netzwerks. Die heutzutage allgemein gebräuchlichen Computer besitzen jedoch höchstens zwei Prozessoren (die Zentraleinheit und eventuell einen mathematischen Co-Prozessor) und einen Arbeitsspeicher, in dem Daten und

Programme getrennt voneinander gespeichert sind. Die Zentraleinheit verarbeitet gemäß den Befehlen des Programms sequentiell die Daten. Rechner mit dieser Architektur werden nach deren Entwickler „von-Neumann-Rechner“ genannt.

Will man sich den extrem teuren Schritt der Hardware-Implementation eines neuronalen Netzes ersparen, besteht die erste Aufgabe einer Computersimulation also darin, mittels der Software ein neuronales Netz zu emulieren. Dies bedeutet, daß im Arbeitsspeicher des Computers Einheiten bereitgestellt werden, die einen den Neuronen eines Netzwerks äquivalenten Zustand einnehmen. Aufgrund der sequentiellen Arbeitsweise der von-Neumann-Rechner können aber nur die Zustände, nicht jedoch die Prozesse eines Netzwerks emuliert werden. Das parallele Verarbeiten der Eingabevektoren in einem neuronalen Netz wird ersetzt durch ein sequentielles, das heißt Neuron für Neuron, Verbindung für Verbindung werden nacheinander abgearbeitet. Dies macht die Computeremulation von neuronalen Netzwerken zu einer sehr zeit- und rechenintensiven Arbeit.

It is sometimes not appreciated by the public at large, including most scientists, how difficult and tricky it can be to get a computer simulation to work. The response to a correctly running simulation is often something along in the lines of "It took you six months just to do that?" With "real" experiments in neuroscience or psychology, there are usually a large number of dead animals or a depleted subject account to show that the experimenter has, indeed, been working. With failed simulations, there is merely a bill for CPU charges, frustration, and the problem of explaining to others why you have been wasting your time. (Anderson & Rosenfeld 1988, 65).

Die Rechenzeit, die für ein neuronales Netz gebraucht wird, hängt von der Rechengeschwindigkeit des Computers und der Qualität des Programms ab. Die Geschwindigkeit der Implementation eines Netzwerks wird in Verknüpfungen pro Sekunde gemessen. Die Höchstzahl der möglichen Verknüpfungen wird durch den Arbeitsspeicher bestimmt. Einen Überblick über die Möglichkeiten heutiger Computer bei effizientester Programmierung und über den geschätzten Bedarf für die Simulation einiger Anwendungsgebiete der menschlichen Informationsverarbeitung bietet Abbildung 15. In einem logarithmischen Koordinatensystem sind auf der Abszisse die Anzahl der Verknüpfungen (*connections*), auf

der Ordinate die Verknüpfungsgeschwindigkeit eingetragen. Eine Verknüpfung entspricht dabei einer Verbindung von einem Neuron zu einem anderen, also einem Gewicht. Die grau unterlegten Bereiche sind Schätzungen, welche Kapazität und Geschwindigkeit für einige kognitive Leistungen benötigt werden. So braucht beispielsweise die Sprachverarbeitung 5.000 bis 1.000.000 Verknüpfungen, die in einer Geschwindigkeit von  $10^7$  bis  $10^{10}$  Verknüpfungen pro Sekunde (*connections per second, cps*) erregt werden müssen. Ein handelsüblicher PC/AT (Punkt 1 in der Abbildung) hingegen erreicht eine Kapazität von ungefähr 100.000 Verbindungen und eine Geschwindigkeit von 10.000 cps.

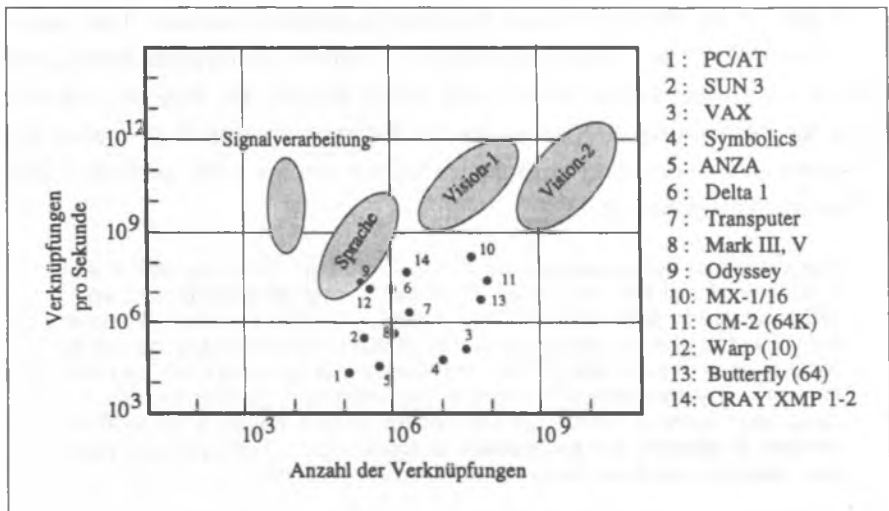


Abb. 15: Geschwindigkeit und Kapazität von Implementationen neuronaler Netzwerke auf verschiedenen Rechenmaschinen (nach Ramacher 1991, übernommen von Rojas 1993, S. 400).

Um die Rechenzeiten für das Netzwerk überschaubar zu halten, müssen die Anzahlen der Eingabeneuronen und der verdeckten Neuronen möglichst gering gehalten werden. Dies ist bei der Suche nach der geeigneten Simulationsaufgabe zu berücksichtigen.

## 4.2 Aufgabe

Auf der Suche nach einem für die Simulation geeigneten Lerneffekt bieten sich die in der letzten Zeit am meisten empirisch untersuchten Gesetzmäßigkeiten motorischen Lernens an: die *variability-of-practice*-Hypothese und der Kontextinterferenz-Effekt.

Schmidt (1975/1990) leitet aus seiner Schematheorie ab, daß variables Üben zum Erwerb der von ihm postulierten Schemata führt, monotones Üben hingegen nicht. Ein Schema oder eine Schemaregel zu erlernen bedeutet, eine Abbildung zu finden, die den Zusammenhang zwischen Ausgangsbedingungen, gewünschtem Bewegungsergebnis und den gewählten Parametern möglichst gut beschreibt. Diese Abbildung wird um so genauer sein, je größer die Anzahl der Punkte im Ergebnisraum ist und je besser sie dort verteilt sind. Eine Verteilung der Punkte im Ergebnisraum auf der Modellebene entspricht dem Variieren von Anfangsbedingungen und gewünschtem Bewegungsergebnis in der Praxis. Je mehr Anfangsbedingungen und Bewegungsergebnis variiert werden, desto besser wird das Schema erlernt. Unter dem Namen „*variability of practice hypothesis*“ (eingeführt von Moxley 1979) ist diese Hypothese als „probably the most important deduction arising from Schmidt's theory“ (vgl. van Rossum 1987, 18) in der Sportwissenschaft bekannt geworden.

Die *variability-of-practice*-Hypothese läßt sich empirisch relativ leicht testen. Zu einem typischen Experiment gehören mindestens zwei Gruppen von Versuchspersonen. Eine Gruppe übt eine Bewegung monoton, eine zweite variabel, wobei nur die Bewegungsparameter, nicht aber die Programminvarianten innerhalb der Übungen variiert werden dürfen. Nach diesen Übungseinheiten wird in einer neuen Aufgabe aus derselben Bewegungsklasse überprüft, ob sich zwischen den Gruppen Unterschiede bei ihrer Bewältigung feststellen lassen. Erwartet wird, daß die variabel übende Gruppe in der Transferaufgabe besser abschneidet.

Die Ergebnisse der empirischen Untersuchungen ergaben – wie so oft in der experimentellen Psychologie – eine uneindeutige Befundlage: Teils

wurden die Vorhersagen von Schmidt bestätigt, teilweise konnten sie nicht nachgewiesen werden (für einen aktuellen Überblick über die empirischen Untersuchungen vgl. van Rossum 1990). Es zeigte sich aber, daß offensichtlich die Reihung der variablen Übungen signifikanten Einfluß auf das Lernen hat. Dies legt nahe, daß die experimentell beobachteten Phänomene nicht auf einer Schemaregelbildung, sondern auf Kontextinterferenz-Effekten beruhen.

Sollen in einer Übungseinheit zwei oder mehr verschiedene Aufgaben gelernt werden, besteht zum einen die Möglichkeit, die Übungseinheit in so viele Übungsblöcke wie Aufgaben zu unterteilen und der Reihe nach die einzelnen Blöcke abzuarbeiten, zum anderen, die verschiedenen Aufgaben hintereinander oder in zufälliger Reihenfolge durcheinander zu lehren. Im ersteren Fall beeinflussen sich die verschiedenen Aufgaben wenig, man spricht von einer geringen Kontextinterferenz, im letzteren wird von einer großen Beeinflussung, einer hohen Kontextinterferenz ausgegangen. Kontextinterferenz ist also ein Maß für die Beeinflussung durch den Übungszusammenhang, für die Interferenz im Übungskontext.

Es konnte nun empirisch nachgewiesen werden, daß die Leistung von Gruppen mit hoher Kontextinterferenz in der Übungsphase im Vergleich zu Gruppen mit niedriger Kontextinterferenz zwar während und direkt nach der Übung erwartungsgemäß niedriger war, sie in einer späteren Leistungsüberprüfung, dem Behaltenstest, sowie in einem Transfertest jedoch besser abschnitten. Dieser Befund ging als *Kontextinterferenz-Effekt (contextual interference effect)* in die wissenschaftliche Literatur ein (z.B. Battig 1979; Shea & Morgan 1979). Magill und Hall fassen in ihrem Überblick über die Forschungen zum Kontextinterferenz-Effekt zusammen, „that the contextual interference effect is a robust effect that can be demonstrated in a variety of motor skill learning conditions“ (1990, 265).

Ein typischer Test zur Überprüfung des Kontextinterferenz-Effekts ist das Experiment von Shea und Morgan (1979), die erstmals die Gültigkeit des Effekts beim motorischen Lernen untersuchten. Dabei mußten die Versuchspersonen drei von sechs Barrieren in einer bestimmten

Reihenfolge in möglichst kurzer Zeit umstoßen. Drei verschiedene Bewegungsmuster wurden geübt, wobei eine Versuchspersonengruppe (Blockgruppe) je einen Übungsblock mit 18 Versuchen pro Bewegungsmuster hatte, die zweite (Zufallgruppe) drei Übungsblöcke mit 18 Versuchen, wobei alle drei Bewegungsmuster sechsmal im Übungsblock in unsystematischer Reihenfolge vertreten waren. War die Blockgruppe während der Aneignungsphase noch im Mittel schneller als die Zufallgruppe, so kehrte sich dieses Ergebnis sowohl für den Behaltenstest als auch für den Transfertest (drei andere Barrieren bzw. fünf Barrieren umstoßen) um.

Sowohl die durch die *variability-of-practice*-Hypothese vorhergesagte Lerngesetzmäßigkeit als auch der Kontextinterferenz-Effekt lassen sich prinzipiell mit einem neuronalen Netzwerk simulieren. Dem Netz können variable und monotone Eingabemuster eingespeist werden, und auch die Reihenfolge der variablen Muster läßt sich variieren.

Eine Simulation eines Experiments zur Überprüfung der *variability-of-practice*-Hypothese soll hier nicht durchgeführt werden, denn ihr Ergebnis läßt sich theoretisch vorhersagen. Angenommen, zwei neuronale Netze sollen eine beliebige Abbildung  $f(\mathbf{x}) = \mathbf{y}$  von Ein- auf Ausgabevektoren erlernen. Das Netz M bekommt monoton einen Eingabevektor  $\mathbf{x}_m$  präsentiert, das Netz V variable Eingabevektoren  $\mathbf{x}_1$  bis  $\mathbf{x}_n$ . Beide Netze werden anfangs einen Fehler produzieren, der zurückgemeldet und im Lernverlauf minimiert wird. Netz M lernt nun (recht schnell) die korrekte Zuordnung von  $\mathbf{x}_m$  zu dem dazugehörigen Ausgabevektor  $\mathbf{y}_m$ . Netz V lernt die Zuordnungen von  $\mathbf{x}_i$  zu  $\mathbf{y}_i$ , wobei – als Eigenschaft der neuronalen Netze – gleichzeitig ein Zusammenhang von  $\mathbf{x}$  und  $\mathbf{y}$  mitgelernt wird. Wird nun, analog zu den Experimenten zum menschlichen Lernen, ein Behaltens- und ein Transfertest durchgeführt, erfüllen sich die Vorhersagen der *variability-of-practice*-Hypothese perfekt. Das monoton trainierte Netz M kann nur die Abbildung von  $\mathbf{x}_m$  auf  $\mathbf{y}_m$  reproduzieren, für einen beliebigen anderen Eingabevektor sind die Ausgaben völlig undefiniert. Mit Schmidt gesprochen, konnte Netz M keine Schemaregel entwickeln. Das variabel trainierte Netz V hingegen wird nicht nur auf

die zuvor im Training präsentierte Trainingsvektoren die richtigen Ausgabevektoren finden, sondern auch auf unbekannte (Transfer-)Eingaben eine sinnvolle Ausgabe erzeugen. Die von Schmidt vorhergesehene Überlegenheit variablen Übens ergibt sich also bei einer Modellierung mit neuronalen Netzwerken trivialerweise aus deren Eigenschaften.

Der Effekt, den verschiedene Kontextinterferenz-Bedingungen auf das Lernen im Netz haben, ist bisher zweimal getestet worden (Masson 1990; Horak 1992). Beide Forscher verwenden ein mehrschichtiges Netzwerk mit einer verdeckten Schicht und *error-backpropagation* als Lernverfahren.

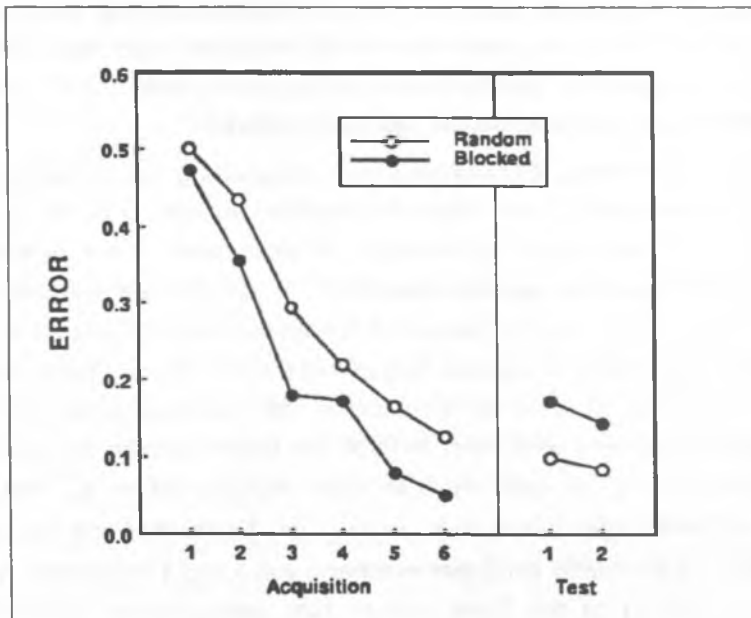


Abb. 16: Mittlerer Fehler zwischen dem Aktivierungsmuster der Ausgabeinheiten und dem zu lernenden Muster über die Aneignungsphase und die Testphase (aus Masson 1990, 234).

Massons Netz besteht aus 6 Eingabeneuronen, 4 Ausgabeneuronen und 12 Neuronen in der verdeckten Schicht. Lernaufgabe ist es, zwei ver-



schiedene abstrakte Mustersequenzen à 4 Muster auf die dazugehörigen Ausgabemuster richtig abzubilden.

Ein Eingabemuster besteht dabei aus einem sechsdimensionalen Eingabevektor, dessen Komponenten entweder 1 oder 0 sind. Bei der Übungsbedingung mit hoher Kontextinterferenz werden die zwei Mustersequenzen abwechselnd je 15mal in das Netz eingegeben, bei niedriger Kontextinterferenz wird 15mal die eine, dann 15mal die andere Sequenz hintereinander gelernt. Im Retentionstest werden die beiden Sequenzen viermal abwechselnd präsentiert. Abbildung 16 zeigt Massons Ergebnisse.

Zusammenfassend kommt Masson (1990, 233) zu dem Schluß, daß

*the pattern of error values followed the empirical data in all critical respects. The improvement in performance during training was greater for the blocked condition, but performance during the retention test was better in the random condition.*

In der Simulation von Horak besteht die Trainingsaufgabe des Netzes darin, für eine eingegebene Distanz die richtige Kraft als Ausgabe zu lernen. Dafür wählt er verschiedene Netzwerke mit einem Eingabe-, einem Ausgabe- und ein bis drei verdeckten Neuronen. Die Ergebnisse werden mit zunehmender Neuronenzahl in der verdeckten Schicht besser. Alle veröffentlichten Versuche zur Kontextabhängigkeit des Lernens beziehen sich auf das Netz mit drei verdeckten Neuronen.

Horak nimmt die Eingabedaten aus drei gleich großen Intervallen, die kurze, mittlere oder lange Distanzen darstellen. Eine Übungseinheit besteht aus sechs Eingabe/Ausgabe-Paaren eines Intervalls, die je dreimal trainiert werden. Insgesamt werden 600 Übungseinheiten absolviert. Das Netz unter geblockten Übungsbedingungen trainiert zuerst 200 Einheiten aus dem kurzen, dann 200 aus dem mittleren und 200 aus dem langen Distanzintervall; das Netz mit sequentiellen Übungsbedingungen trainiert 600 Einheiten abwechselnd kurz, mittel, lang; ein drittes Netz trainiert 600 Einheiten in zufälliger Abfolge.

Horaks Ergebnisse entsprechen denen Massons. Das Netz mit geblockten Trainingsbedingungen lernt deutlich schneller als die beiden anderen,

aber in einem Transfertest (mit einer nicht in der Trainingsmenge enthaltenen Weite) schneiden sowohl das Netz mit sequentiellen als auch das mit zufälligen Übungsbedingungen weit besser ab.

Bei der Suche nach einer geeigneten, möglichst sporttypischen Bewegungsaufgabe, die in der Simulation die Netzeingabe und -ausgabe festlegt, muß an die oben dargelegten Beschränkungen erinnert werden, dabei sind die Einzelheiten der Netzarchitektur zu berücksichtigen. Die Modellierung des Kontextinterferenz-Effekts kann selbstverständlich nur mit einem Netzmodell geschehen, das zielorientiert lernen kann. Daher bietet sich hier vor allem eine Emulation von Jordans oben beschriebenen Netzwerk (Abb. 11, S. 67) an.

Die Kapazität eines neuronalen Netzes und damit die mögliche Komplexität der simulierten Aufgabe hängen von der Anzahl der Neuronen und ihrer Verbindungen ab. In ihrer Simulation von dynamischen Bewegungen benutzen Jordan und Rumelhart (1992) ein Netz mit jeweils 50 Neuronen in den verdeckten Schichten. Damit liegt ihre Simulation sowohl bezüglich des Arbeitsspeichers als auch der Rechenzeit weit jenseits der Grenze dessen, was auf einem für die hier geplante Simulation zur Verfügung stehenden „Personal Computer“ (PC/AT, vgl. Abb. 15) möglich wäre.<sup>8</sup>

In Anlehnung an Jordan & Rumelhart liegt es nahe, das Netz für gewünschte Bewegungsergebnisse zugehörige Bewegungsvektoren lernen zu lassen, die dann in Interaktion mit den Umgebungsbedingungen tatsächliche Bewegungsergebnisse zur Folge haben. Dieses Modell muß jedoch aus Gründen der Beschränkungen von Rechenzeit und Speicherkapazität erheblich eingeschränkt werden.

Als erste Einschränkung bietet sich an, eine Simulationsaufgabe zu wählen, bei der der Einfluß der Umwelt konstant ist, also wechselnde Umgebungsbedingungen keine Rolle spielen. Das Ergebnis der Bewegung wird dann direkt aus dem Bewegungsvektor errechnet.

---

<sup>8</sup> Leider berichtet Jordan nie, auf welcher Rechenmaschine seine Simulationen liefen und wieviel Rechenzeit sie benötigten.

Zweitens muß der Bewegungsvektor selbst vereinfacht werden. Um die Schwierigkeiten anatomischer Modellbildung zu umgehen, wird statt dessen die Modellierung der Bewegung aus beobachtbaren biomechanischen Parametern gewählt. Neben der Dimensionsreduktion des Bewegungsvektors hat dies den Vorteil, daß – ein korrektes biomechanisches Modell vorausgesetzt – direkt aus diesen Parametern das Ergebnis der Bewegung errechnet werden kann. Man könnte einwenden, daß dieser Ebenenwechsel (von der Ebene der Muskelaktivierung, in dem die Bewegungsvektoren abgebildet werden, zur Ebene äußerlich beobachtbarer biomechanischer Parameter) die Simulation entwertet. Dies trifft jedoch nicht zu. Für die Aktivierung der Neuronen und das Lernen im Netz spielt die Ebene, auf der die Vektoren in der realen Welt abgebildet sind, keine Rolle, das richtige Einstellen der Gewichte erfolgt nach denselben Prinzipien. Grundsätzlich wäre es auch leicht möglich, ein Netz zu trainieren, das die Abbildung zwischen Muskelaktivierungsvektoren und biomechanischen Vektoren lernt.

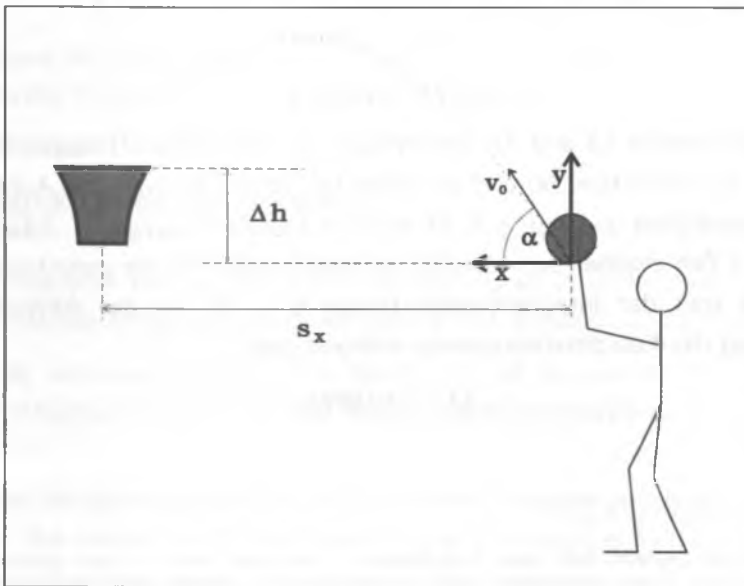


Abb. 17: Simulationsaufgabe.

Eine hervorstechende Eigenschaft neuronaler Netze ist, daß sie komplexe nichtlineare Zusammenhänge erkennen können. Um die Simulation nicht trivial zu gestalten, sollte die Ausgabe nichtlinear von den Eingabeparametern abhängen. Damit bietet sich die Simulation einer Wurfpabel an.

Unter Berücksichtigung obiger Überlegungen eignet sich als Simulationaufgabe der Positionswurf im Basketball. Das gewünschte Bewegungsergebnis ist der Korberfolg, die Eingabe in das Netz ist daher die Entfernung zum Korb. Das Netz soll den Abflugwinkel und die Abfluggeschwindigkeit lernen, die nötig sind, um in den Korb zu treffen. Dabei wird von dem in Abbildung 17 veranschaulichten zweidimensionalen Modell ausgegangen.

Aus den Bewegungsgleichungen kann der Zusammenhang zwischen Abflugwinkel und -geschwindigkeit und der Entfernung des Korbs errechnet werden. Der Luftwiderstand spielt bei den geringen Geschwindigkeiten und dem Gewicht des Basketballs nur eine unbedeutende Rolle und wird im folgenden vernachlässigt<sup>9</sup>.

$$v_x(t) = v_0 \cos \alpha \quad (13)$$

$$v_y(t) = v_0 \sin \alpha - g t \quad (14)$$

Die Gleichungen 13 und 14 beschreiben die momentane Fluggeschwindigkeit in horizontaler ( $x$ ) und vertikaler ( $y$ ) Richtung (mit  $v_0$  als Anfangsgeschwindigkeit und  $|g| = 9,81 \text{ ms}^{-2}$  als Erdbeschleunigung). Integriert über die Zeit ergeben sich daraus die Gleichungen für die zurückgelegte Strecke (mit der Integrationskonstanten  $s_0 = 0$ , da der Abwurf im Ursprung des Koordinatensystems erfolgen soll):

$$s_x(t) = v_0 t \cos \alpha \quad (15)$$

<sup>9</sup>

Die anfangs geplante Idee, einen Fußballabstoß zu simulieren (wobei zu der gewünschten Entfernung Abflugwinkel und -geschwindigkeit zu ermitteln wären) und dabei den Luftwiderstand zu berücksichtigen, wäre dadurch uninteressant geworden, daß erstaunlicherweise gerade durch den Luftwiderstand eine fast lineare Beziehung zwischen Abfluggeschwindigkeit und Weite besteht (vgl. Daish 1972, 52).

$$s_y(t) = v_0 t \sin \alpha - \frac{1}{2} g t^2 \quad (16)$$

Der Ball erreicht die Höhe des Korbs bei  $s_y = \Delta h$  (mit  $\Delta h$  als Höhendifferenz zwischen Abflugpunkt und Ring). Um die Flugzeit zu berechnen, die der Ball braucht, um auf Korbhöhe zu sein, wird  $\Delta h$  für  $s_y$  in Gleichung 16 eingesetzt und nach  $t$  aufgelöst. Da nur interessiert, wann der Ball von oben durch den Ring fliegt, reicht es, nur den größeren Funktionswert zu betrachten.

$$s_y = \Delta h \Rightarrow t_{Korb} = \frac{v_0 \sin \alpha + \sqrt{(v_0 \sin \alpha)^2 - 2 \Delta h g}}{g} \quad (17)$$

Diese Zeit  $t_{Korb}$  wird jetzt in Gleichung 15 eingesetzt und man erhält eine Gleichung für die horizontale Entfernung des Korbs von dem Abflugpunkt in Abhängigkeit von Abflugwinkel und -geschwindigkeit (und konstantem  $\Delta h$ ).

$$s_{x_{Korb}}(v_0, \alpha) = \frac{v_0 \cos \alpha}{g} \left( v_0 \sin \alpha + \sqrt{v_0^2 \sin^2 \alpha - 2 \Delta h g} \right) \quad (18)$$

Mit dieser Aufgabe soll nun zunächst überprüft werden, ob der Kontextinterferenz-Effekt auch für das Lernen im Modell gilt.

### 4.3 Struktur des Programms

Das Programm für die Emulation des Netzwerks in dieser Arbeit ist in vier Einheiten unterteilt (der Quellcode ist dem Anhang zu entnehmen).

- In der untersten Einheit („unit datastru“), auf die alle anderen Einheiten zurückgreifen, werden die nötigen *Datenstrukturen* zur Verfügung gestellt.
- In der darüberliegenden Einheit („unit data“) werden die *Daten* verwaltet. Hier werden die Zufallsdaten für das Training des Bewegungsvorstellungsnetzes erzeugt und in einem Feld gespeichert, die einzelnen Eingabe- und Ausgabevektoren werden normiert bzw. entnormiert.

- In einer dritten Einheit („unit nets“) geschieht die eigentliche Emulation der Netze. Je nach Anforderung aus dem Hauptprogramm werden die verschiedenen Netze (Bewegungsvorstellungsnetz, Bewegungssteuerungsnetz) mit Zufallsgewichten zwischen -0,5 und 0,5 initialisiert. Die Prozedur „feedforward“ propagiert den aus der „unit data“ erhaltenen Eingabevektor durch das Netz. Dazu werden mit der Prozedur „pass“ der Erregungsvektor mit dem Gewichtevektor multipliziert, die Funktion „fire“ liefert dann die entstehende Erregung der einzelnen Neuronen. Mit der Prozedur „compute\_delta2“ wird der Fehler, also die Differenz zwischen der Erregung und dem normierten Sollwert, zurückgeführt und dabei  $\delta_2$ , die partielle Ableitung des Fehlers nach den mit den Ausgabeneuronen verbundenen Gewichten, berechnet. Analog dazu berechnet „compute\_delta1“ die partielle Ableitung des Fehlers nach den Gewichten der Verbindungen mit den Neuronen der verdeckten Schicht ( $\delta_1$ ). Die Prozedur „correction“ verändert dann die Gewichte in Richtung des negativen Gradienten ( $-\delta_2$  bzw.  $-\delta_1$ ), multipliziert mit der Konstanten  $\gamma$ . Koordiniert werden diese Befehle durch die Prozeduren „lernen1“ bzw. „lernen2“; „lernen1“ ist für das Training des Bewegungsvorstellungsnetzes zuständig, „lernen2“ regelt das Training des Bewegungssteuerungsnetzes. Zusätzlich wird in der „unit nets“ das Speichern und Laden der Netze auf externe Speicher sichergestellt.
- Im Hauptprogramm „program connectionist\_model“ wird die eigentliche *Aufgabenstellung* bearbeitet. Die Prozedur „learn\_forward\_model“ steuert das Lernen des Bewegungsvorstellungsnetzes, „test\_forward\_model“ testet das erlernte Bewegungsvorstellungsnetz mit zufälligen Eingabevektoren, „learn\_inverse“ legt die Konfiguration des Netzwerks für das Erlernen der Bewegungssteuerung fest, benennt die Protokolldateien und legt die Gruppengröße fest, die Prozedur „contextual\_interference“ legt die Übungsbedingungen fest.

Das Lernen des Bewegungsvorstellungsnetzes dauert auf einem PC mit dem Prozessor iAPX 80486, einem dazugehörigen mathematischen Co-Prozessor und einer Taktfrequenz von 33 MHz ungefähr fünf Stunden,

d. h. eine Iteration mit 500 Trainingsvektoren dauert ungefähr 9 Sekunden. Das Problem bei einer solchen Rechenzeit ist, daß anfangs unvermeidlich vorhandene Programmierfehler u. U. erst nach fünf Stunden bemerkbar werden, was das „Entlausen“ („Debuggen“) des Programms sehr zeitraubend macht.

#### 4.4 Das Training des Bewegungsvorstellungsnetzes

Zuerst wird das Bewegungsvorstellungsnetz trainiert. Eine zufällige Auswahl von 500 Kombinationen von Abflugwinkeln zwischen 40 und 60 Grad und -geschwindigkeiten zwischen 7 und 11 m/s werden dem Netz als Trainingsvektoren eingegeben, der Ergebniswert berechnet sich nach Gleichung 18 mit  $\Delta h = 1$ . Alle dem Netz eingegebenen Werte müssen normiert werden, da nur Werte zwischen Null und Eins ausge-

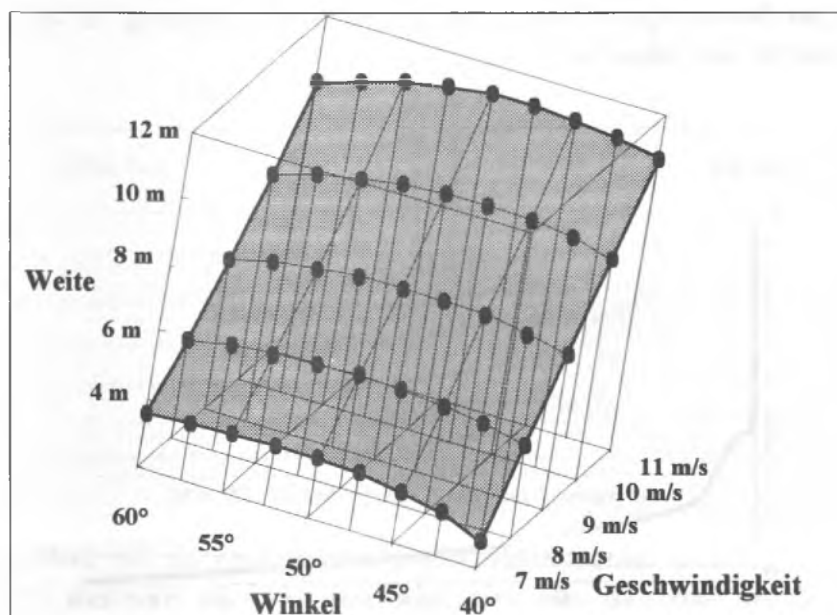


Abb. 18: Graphische Veranschaulichung von Gleichung 18.

geben werden können und die Eingabe in das Bewegungsvorstellungsnetz im weiteren Verlauf der Simulation die Ausgabe des Bewegungssteuerungsnetzes sein wird. Da Null und Eins die Grenzwerte der Sigmoiden für Erregungen zwischen  $-\infty$  bzw.  $\infty$  bilden (vgl. Abb. 8, S. 44), sie also nur bei einer „unendlichen“ Erregung eines Neurons erreicht werden, ist es sinnvoll, die Normierungsfunktion so zu wählen, daß ihre Funktionswerte für die Trainingsdaten zwischen 0,1 und 0,9 liegen.

Das Bewegungsvorstellungsnetz hat zwei Eingabeneuronen. An dem ersten liegt die Abfluggeschwindigkeit an, am zweiten der Abflugwinkel. Die verdeckte Schicht besteht aus fünf Neuronen, die Ausgangsschicht aus einem Neuron. Gelernt wird mit dem üblichen *backpropagation*-Algorithmus, die Lernkonstante  $\gamma$  hat den Wert 1, ebenso die Temperaturkonstante  $c$ . Die zu lernende Funktion aus Gleichung 18 wird durch Abbildung 18 veranschaulicht.

Nach 2000 Iterationen mit je 500 Trainingsvektoren wird die Lernphase für das Bewegungsvorstellungsnetz abgebrochen. Abbildung 19 veranschaulicht den Lernverlauf.

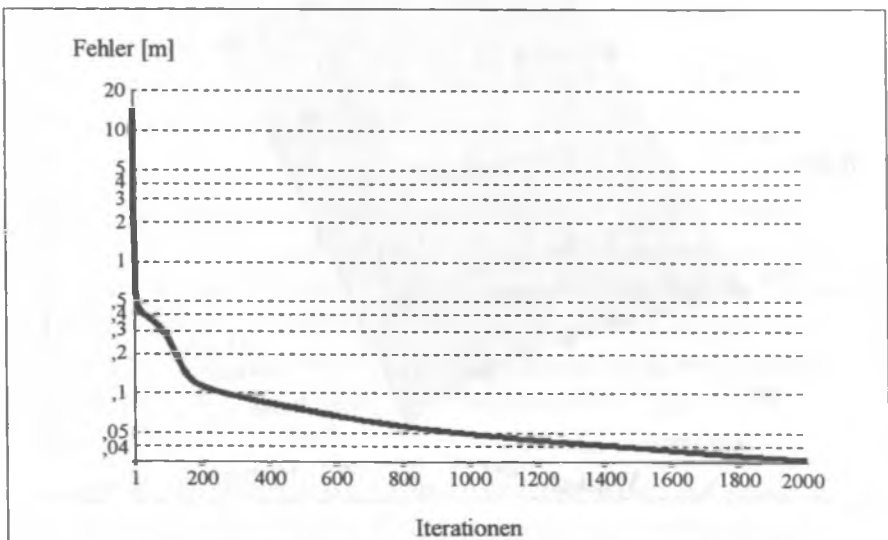


Abb. 19: Verlauf des Gesamtfehlers (in Metern) über die Iterationen.



Über den Iterationen auf der Abszisse wird der Gesamtfehler aufgetragen, der sich als Summe der quadrierten Abstände zwischen der unnormierten Netzausgabe und dem errechneten Ergebnis über alle 500 Trainingsvektoren ergibt.

In einem Test werden 100 zufällig ausgewählte Eingabevektoren in das Netz eingegeben. Der durchschnittliche Abstand zwischen der unnormierten Netzausgabe und dem nach Gleichung 18 errechneten Ergebnis beträgt 0,07 m, die Standardabweichung 0,07 m, die Verteilung der Fehler über Winkel und Geschwindigkeit veranschaulicht Abbildung 20. Dabei ist zu erkennen, daß vor allem bei Winkeln und Geschwindigkeiten am Rand des Eingabebereichs die Fehler noch relativ hoch sind, während der Fehler in der Mitte nahezu Null beträgt.

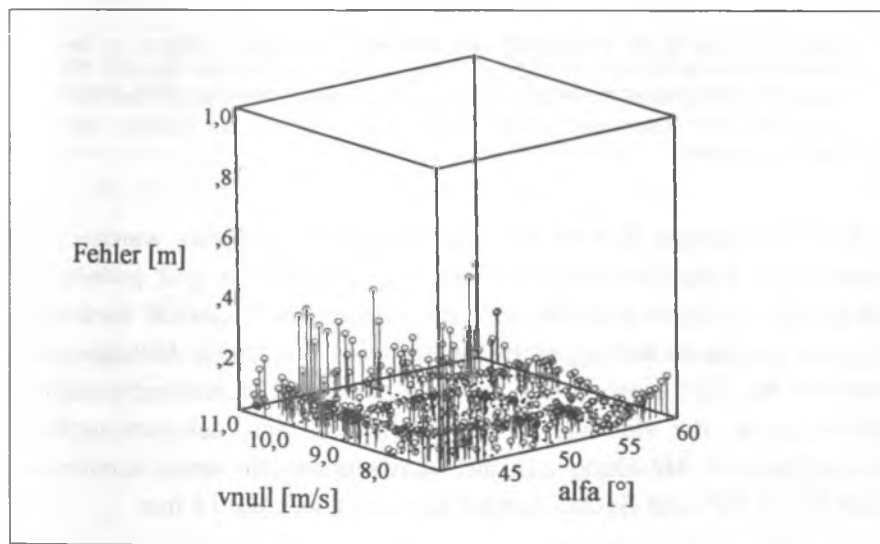
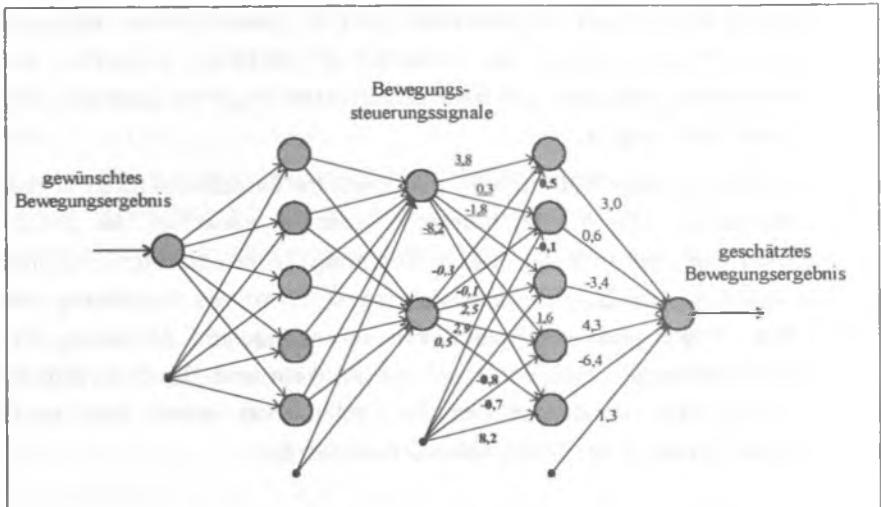


Abb. 20: Verteilung der Fehler der Testvektoren.

Ein Netz mit so niedrigdimensionalen Vektorräumen läßt sich noch relativ leicht ein wenig genauer analysieren. Die Gewichte des Bewegungsvorstellungsnetzes sind aus Abbildung 21 zu entnehmen.



**Abb. 21:** Das in der Simulation benutzte Netz. Das obere Neuron in der dritten Schicht codiert die Geschwindigkeit, das untere den Winkel. Die Gewichte nach dem Training des Bewegungsvorstellungsnetzes sind eingetragen. Die schwarzen Punkte feuern jeweils eine 1, sie ersetzen den Schwellenwert.

Um herauszufinden, wie die einzelnen Neuronen reagieren, werden zwei Reihen von Eingabevektoren in das Netz eingespeist und jeweils ein Parameter konstant gehalten und der andere kontinuierlich verändert. Für eine konstante Abfluggeschwindigkeit von 8 m/s und Abflugwinkeln von  $40^\circ$  bis  $60^\circ$  ergeben sich die in Abbildung 22a veranschaulichten Aktivierungen der Neuronen der mittleren Schicht, dementsprechend veranschaulicht Abbildung 22b die Aktivierungen für einen konstanten Winkel von  $50^\circ$  und Geschwindigkeiten von 7 m/s bis 11 m/s.

Aus den Abbildungen ist zu erkennen, daß der Abflugwinkel von dem dritten und dem vierten Neuron der verdeckten Schicht kodiert wird, die Abfluggeschwindigkeit durch ein Zusammenwirken der Neuronen eins, drei und fünf. Dabei muß festgehalten werden, daß diese Konfiguration des Netzes abhängig ist von seinem zufälligen Anfangszustand.

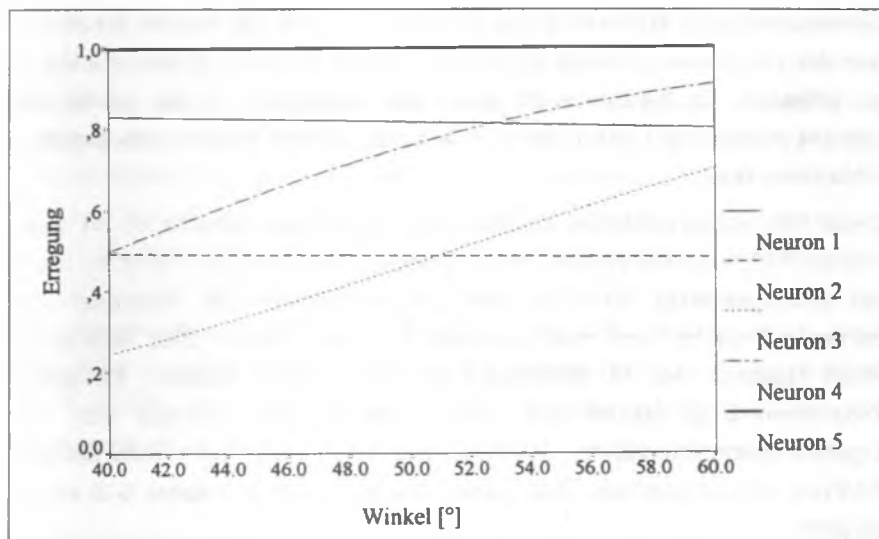


Abb. 22a: Aktivierung der Neuronen der verdeckten Schicht bei einer konstanten Abfluggeschwindigkeit von 8 m/s in Abhängigkeit vom Abflugwinkel.

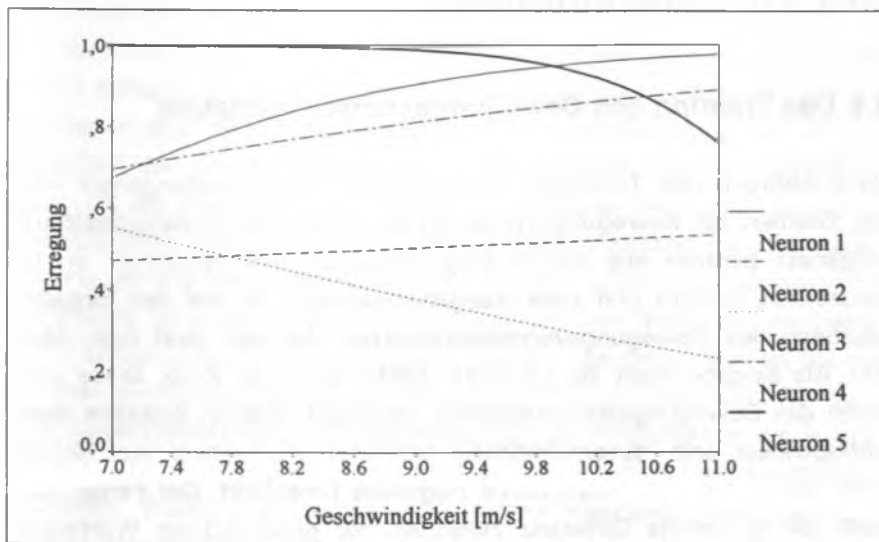


Abb. 22b: Aktivierung der Neuronen der verdeckten Schicht bei einem konstanten Abflugwinkel von 55° in Abhängigkeit von der Abfluggeschwindigkeit.

Generalisierbares Ergebnis dieser Analyse ist, daß die Anzahl der Neuronen der verdeckten Schicht ausreicht, um das gestellte Problem adäquat zu erfassen. Es hätten wohl auch vier Neuronen in der verdeckten Schicht ausgereicht, da offensichtlich das zweite Neuron das Ergebnis nicht beeinflußt.

Diese hier vorgenommene Analyse der verdeckten Schicht ist im allgemeinen Fall, also bei Netzen mit mehreren Neuronen pro Schicht, weniger aussagekräftig, da sich dann die Aktivitäten der Neuronen der mittleren Schicht nicht mehr sinnvoll zuordnen lassen. Dies beleuchtet einen Nachteil der Modellbildung mit neuronalen Netzen: Komplexe Zusammenhänge lassen sich zwar erlernen, aber woraus sich das Ergebnis zusammensetzt, ist u. U. nur nach einer sehr aufwendigen Analyse nachvollziehbar. Auf diesen Aspekt wird in Kapitel 5.3 eingegangen.

Insgesamt kann festgestellt werden, daß ein Bewegungsvorstellungnetz mit fünf Neuronen in der verdeckten Schicht die in der Simulation gewählte Lernaufgabe gut bewältigt.

## 4.5 Das Training des Bewegungssteuerungsnetzes

Nach Abbruch des Trainings für das Bewegungsvorstellungnetz wird das Erlernen der Bewegungssteuerung simuliert. Das Bewegungssteuerungsnetz besteht aus einem Eingabeneuron, fünf Neuronen in der verdeckten Schicht und zwei Ausgabeneuronen, die mit den Eingabeneuronen des Bewegungsvorstellungnetzes identisch sind (vgl. Abb. 21). Als Eingabe dient die normierte Entfernung zum Korb. Diese wird durch das Bewegungssteuerungsnetz geschickt, dessen Ausgabe dann Abflugwinkel und -geschwindigkeit darstellen. Außerhalb des Netzes wird daraus dann die tatsächliche Flugweite berechnet. Der Fehler, das heißt die quadrierte Differenz zwischen der gewünschten Wurfweite und der berechneten tatsächlichen Wurfweite, wird durch das Bewegungsvorstellungnetz zurückgeführt. Dabei bleiben die Gewichte im Bewegungsvorstellungnetz unverändert. Der Fehler wird weiter fortge-

schrieben, bis in der dritten Schicht, der Schnittstelle zwischen Bewegungsvorstellungs- und Bewegungssteuerungsnetz, der jeweilige Anteil des Fehlers an den Ausgabeneuronen des Bewegungssteuerungsnetzes angelangt ist. Mathematisch gesehen ist dies die partielle Ableitung der Fehlerfunktion nach den Gewichten der Ausgabeneuronen. Hier werden die Gewichte verändert und der Fehler bis zur Eingabeschicht weiterpropagiert. Zu beachten ist hierbei, daß wegen der Fehlerrückführung durch das Bewegungsvorstellungsnetz die Summe der Fehleranteile nicht mehr dem Gesamtfehler entspricht, sondern wesentlich kleiner wird. Dies muß dadurch ausgeglichen werden, daß die Lernkonstante (vgl. Kapitel 2.6.1) erhöht wird. Bei der Simulation in dieser Arbeit wird für das Bewegungssteuerungsnetz eine Lernkonstante von  $\gamma = 5$  verwendet.

#### 4.6 Kontextinterferenz

Zur Überprüfung des Kontextinterferenz-Effekts ist es notwendig, den Lernerfolg des Netzes in Abhängigkeit von dem Grad der Interferenz im Übungskontext zu bestimmen. Lerngeschwindigkeit und Lernerfolg eines Netzes hängen jedoch auch von dem Anfangszustand der Netzgewichte, oder, in geometrischer Analogie, von dem Startpunkt auf der Fehlerlandschaft ab. Diese Einflußgröße ist theoretisch nicht vorhersagbar, daher ist es notwendig, sie durch statistische Verfahren herauszurechnen. Dazu wird die Simulation mit mehreren Netzen pro Übungsbedingung durchgeführt. Netzwerke zu Gruppen zusammenzufassen und Lerneffekte varianzanalytisch festzustellen ist eine Methode, die bisher in der Literatur noch nicht propagiert wurde, allerdings wurden bis auf die oben erwähnten Studien von Horak (1992) und Masson (1990) auch noch keine Lerneffekte in Abhängigkeit von der Präsentation der Vektoren, also unabhängig von der Netzarchitektur und dem Lernalgorithmus, überprüft. Wollte man verschiedene Netzarchitekturen oder Lernalgorithmen miteinander vergleichen, so wären theoretisch mathematische Analysen die adäquate Methode.

#### 4.6.1 Hypothesen

Aufgrund der in der Literatur vorliegenden, oben kurz dargestellten Ergebnisse sowie der Annahme, daß das Lernen im Netz analoge Befunde zum menschlichen Lernen liefert, kommt man, den Kontextinterferenz-Effekt betreffend, zu folgenden Hypothesen:

*H 1: Ein höheres Maß an Interferenz im Übungskontext führt in einem neuronalen Netzwerk zu schlechteren Leistungen während der Lernphase.*

*H 2: Ein höheres Maß an Interferenz im Übungskontext führt zu besseren Leistungen eines neuronalen Netzwerks im Test der Lernleistung.*

*H 3: Ein höheres Maß an Interferenz im Übungskontext führt zu besseren Leistungen eines neuronalen Netzwerks im Test der Transferleistung.*

#### 4.6.2 Design

Die Lernaufgabe besteht darin, zu den vorgegebenen gewünschten Weiten jeweils eine entsprechende Kombination von Abflugwinkel und Abfluggeschwindigkeit zu finden. Dabei werden vier verschiedene Weiten trainiert: drei, fünf, sieben und neun Meter.

Getestet werden sechs Gruppen aus je 20 Netzen. In 800 Iterationen sollen die Netze die Gewichte entsprechend der Lernaufgabe verändern. Die Netzgruppe mit der geringsten Kontextinterferenz (Gruppe 6) übt jede der vier Aufgaben in einem Block mit 200 Iterationen, die Netzgruppe mit der höchsten Kontextinterferenz (Gruppe 2) übt die verschiedenen Aufgaben hintereinander, hat also 800 „Blöcke“ mit jeweils einem Trainingsvektor. Drei weitere Gruppen (3, 4 und 5) ordnen sich dazwischen ein, eine sechste Gruppe von Netzen (Gruppe 1) bekommt die Trainingsvektoren in zufälliger Reihenfolge präsentiert. Tabelle 1 verdeutlicht die unterschiedlichen Übungsbedingungen zwischen den Gruppen.

**Tab. 1: Übersicht über die Übungsbedingungen für die verschiedenen Gruppen von neuronalen Netzen.**

Gruppe	1	2	3	4	5	6
Anzahl der Blöcke	-	800	400	160	80	4
Wiederholungen pro Block	-	1	2	5	10	200
Reihenfolge	zufällig	3-5-7-9	3-5-7-9	3-5-7-9	3-5-7-9	3-5-7-9
Trainingsvektoren (gesamt)	800	800	800	800	800	800

Daten über den Lernverlauf werden zu vier Meßzeitpunkten erhoben: zu Beginn und nach 20, 100 und 180 Iterationen. Dabei werden, nach dem Vorbild von Shea und Morgan (1979), für jeden Meßzeitpunkt die Fehler der verschiedenen Übungsbedingungen zusammengefaßt. Um z.B. den Fehler nach 20 Iterationen zu erhalten, wird bei allen Netzen der Abstand zwischen erreichter und gewünschter Weite bei der 20sten Iteration der jeweiligen Übungsbedingung berechnet. Dieser wird dann über die jeweils 20 Netze einer Gruppe gemittelt. In der Gruppe 2, in der die Netze immer hintereinander 3, 5, 7, und 9 Meter erreichen sollen (800 Blöcke mit einem Versuch pro Block) erhält man die jeweiligen Werte in der 77sten bis 80sten Iteration des Netzes, bei Netzen der Gruppe 6 (vier Blöcke à 200 Versuche) erfaßt man sie im 20sten, 220sten, 420sten und 620sten Durchgang. In der Gruppe 1 muß eine „Strichliste“ geführt werden, wann die 20ste, 100ste und 180ste Wiederholung einer Versuchsbedingung stattfindet, da hier die Reihenfolge zufällig bestimmt wird. Es kann passieren, daß die 800 Iterationen erreicht sind, ohne daß alle Zielweiten 180mal wiederholt wurden.

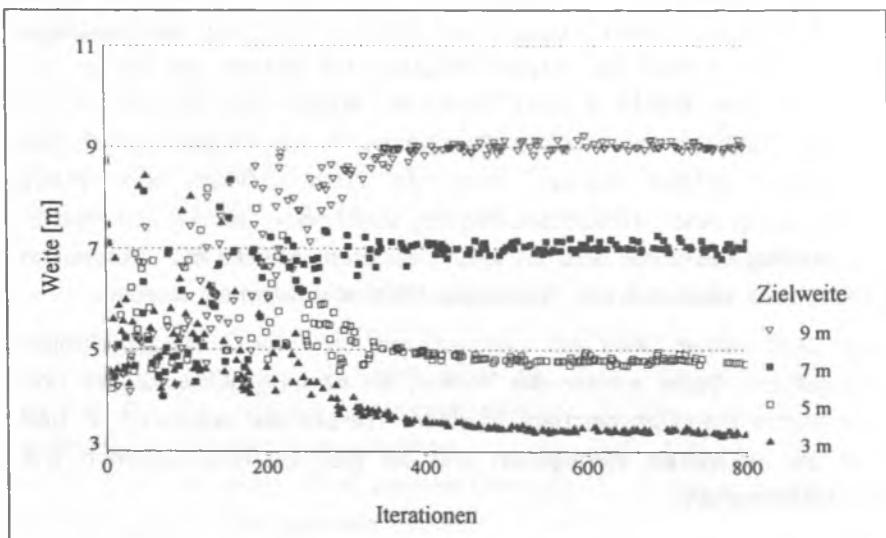
Nach dem Lernen wird ein Lerntest mit den vier Trainingsvektoren durchgeführt. Dabei werden der Winkel, die Abfluggeschwindigkeit und die erreichte Weite gespeichert. In einem Transfertest werden 4, 6, und 10 m als Zielweiten vorgegeben und die gleichen Netzausgaben wie oben festgehalten.

#### 4.6.3 Betrachtung einzelner Lernverläufe

Der Lernverlauf der Netze unter den verschiedenen Kontextinterferenz-Bedingungen soll zunächst an dem Beispiel eines beliebig ausgewählten

Netze pro Gruppe graphisch dargestellt werden (Abb. 23a-f). Dabei entspricht der Abbildung 23a ein Netz aus der Gruppe 1, Abbildung 23f ein Netz der Gruppe 6.

Einige Gesetzmäßigkeiten des Lernens neuronaler Netze können schon aus den Graphiken geschlossen werden. Bei zufälliger Eingabe der Trainingsvektoren (Abb. 23a) ordnen sich nach anfangs chaotischer Ausgabe (bis  $N \approx 200$ ) zwischen den Iterationen 200 und 300 die Ausgaben ihren Zielweiten zu, ab  $N \approx 350$  werden die Zielweiten recht konstant reproduziert. Für die übrigen Netze gilt, daß die aktuell produzierte Weite stark von der vorhergehenden Weite abhängt. In Abbildung 23b z.B. wird deutlich, daß für Zielweiten von 5, 7 und 9 m sich die Ausgabe von unten dem Ziel nähert, während die Zielweite von drei Metern von oben erreicht wird. Dies liegt daran, daß für erstere die vorherige Zielweite niedriger waren, nur bei einer Zielweite von 3 m war mit 9 m das vorherige Ziel weiter.



**Abb. 23a:** Lernverlauf eines Bewegungssteuerungsnetzes der Gruppe 1 über 800 Iterationen. Die Position einer Markierung gibt die tatsächlich erreichte Weite (in Metern) an. Die verschiedenen Markierungen entsprechen den verschiedenen Zielweiten.



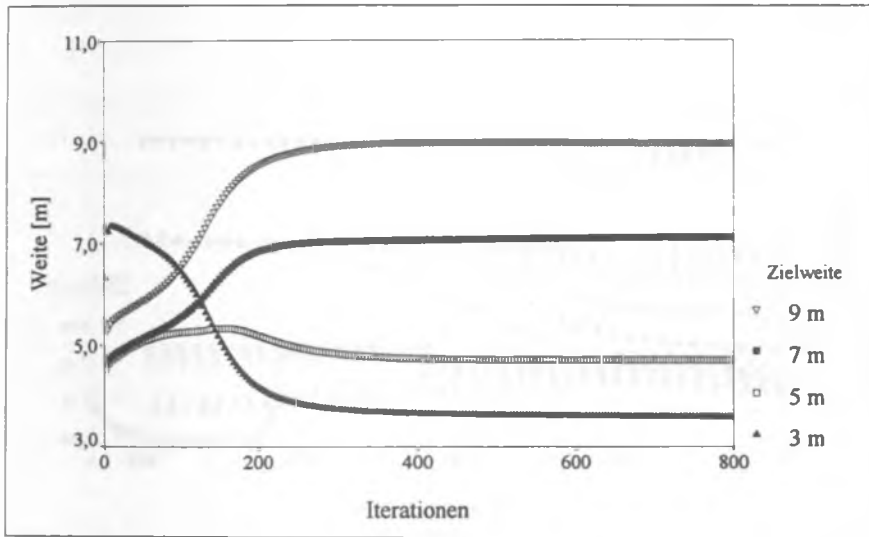


Abb. 23b: Lernverlauf eines Bewegungssteuerungsnetzes der Gruppe 2 über 800 Iterationen (nähere Erläuterungen s. Abb. 23a).

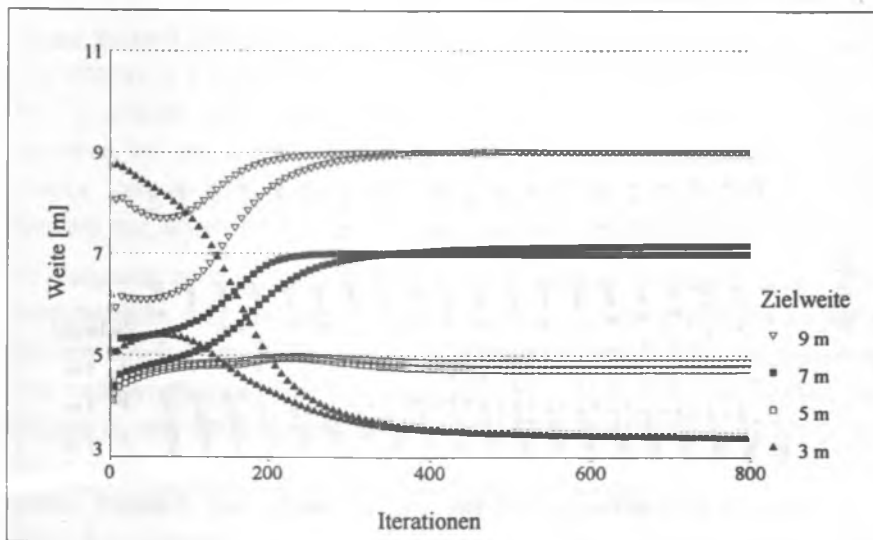


Abb. 23c: Lernverlauf eines Bewegungssteuerungsnetzes der Gruppe 3 über 800 Iterationen (nähere Erläuterungen s. Abb. 23a).

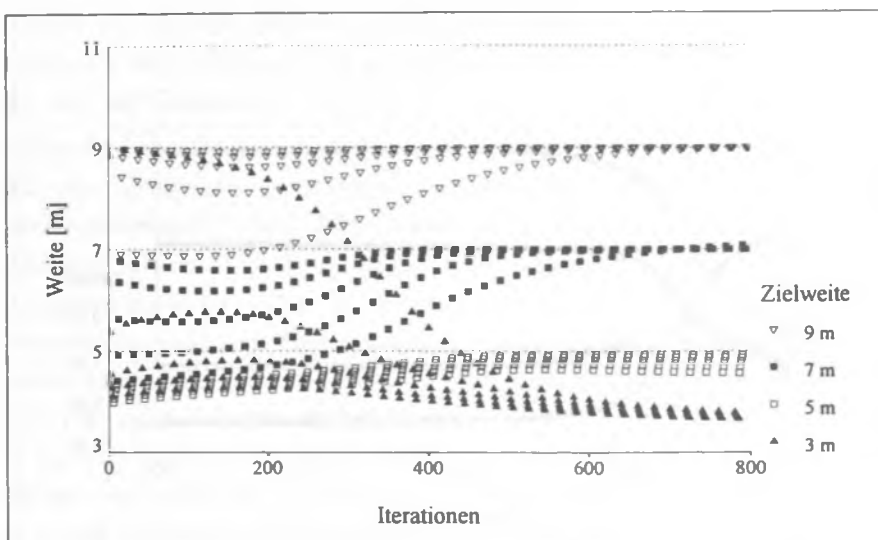


Abb. 23d: Lernverlauf eines Bewegungssteuerungsnetzes der Gruppe 4 über 800 Iterationen (nähere Erläuterungen s. Abb. 23a).

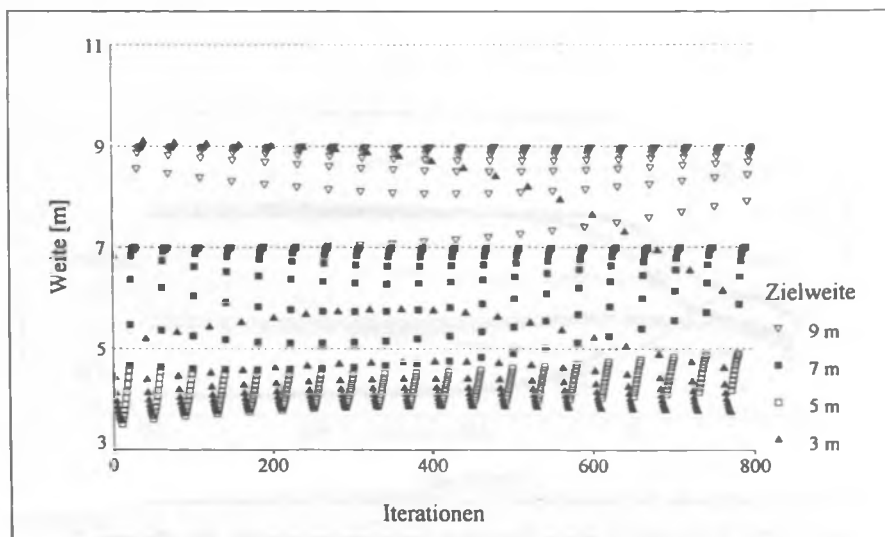


Abb. 23e: Lernverlauf eines Bewegungssteuerungsnetzes der Gruppe 5 über 800 Iterationen (nähere Erläuterungen s. Abb. 23a).

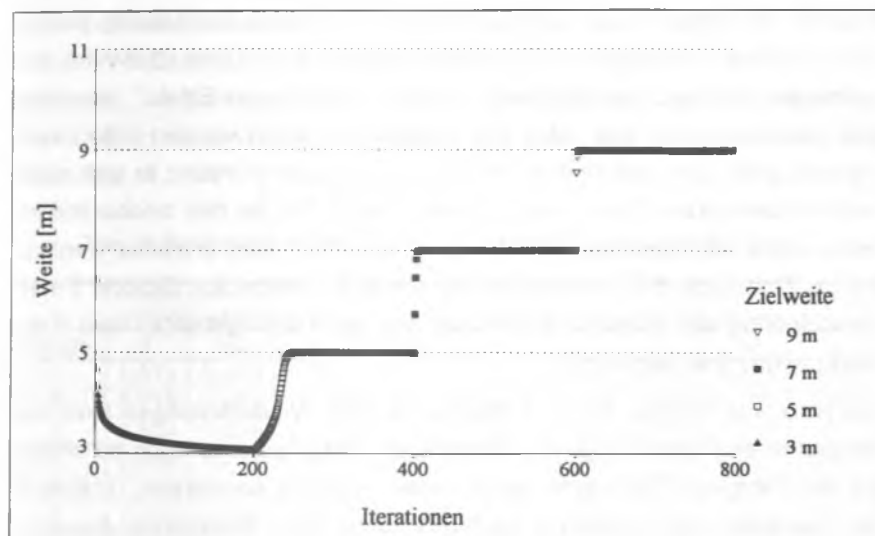


Abb. 23f: Lernverlauf eines Bewegungssteuerungsnetzes der Gruppe 6 über 800 Iterationen (nähere Erläuterungen s. Abb. 23a).

Dieser Befund zieht sich durch alle weiteren Lernverläufe. In Abbildung 23c (Blöcke à 2 Wiederholungen) liegt der erste Versuch noch recht nah am Vorversuch, der zweite dann schon näher an der Zielweite (deutlich zu sehen bei der Zielweite von 3 und 9 m. Welches der erste und der zweite Versuch innerhalb eines Blocks ist, kann nicht der Graphik, sondern nur einem Blick in die Rohdatenliste entnommen werden).

Im Vergleich zu dem Lernverlauf des Netzes aus Gruppe 5 (Abb. 23e) wird deutlich, daß der erste Versuch in einem neuen Block um so näher an dem zuvor geübten Block liegt, je mehr Wiederholungen es innerhalb des vorigen Blocks gab. So liegt der erste Versuch des Netzes der Gruppe 5, die Zielweite von 3 m zu reproduzieren, bis ca. zur 400. Iteration dicht an der 9 m-Marke und ist auch bei der 761. Iteration (dem ersten Versuch des letzten Blocks mit Zielweiten von 3 m) noch nicht unter 6 m gefallen. Dies gilt mit Abstrichen auch für die anderen Zielweiten.

Im Netz der Gruppe 5 werden 7 und 9 m schon im ersten Block perfekt

erreicht, bei Beginn des zweiten Blocks ist diese Einstellung jedoch schon wieder „vergessen“, es werden zunächst 7 m, die Zielweite des vorherigen Blocks, reproduziert. Dieser „Vergessens-Effekt“ ist nach 800 Iterationen für das Netz der Gruppe 3 verschwunden (das beste Ergebnis stellt sich sofort ein), im Netz der Gruppe 5 macht er sich noch deutlich bemerkbar. Falls jedoch dieser Trend für die hier beobachteten Netze keine zufällige Eigenschaft ist, kann schon jetzt erwartet werden, daß im Test nach 800 Iterationen vor allem für Netze aus Gruppe 5 eine Verschiebung der Ergebnisse hin zum letzten Trainingsblock, dem 9 m-Block, erkennbar sein wird.

Das Netz aus Gruppe 6 mit 4 Blöcken à 200 Wiederholungen lernt die Zielweiten am besten und am schnellsten. Dabei scheint auch so etwas wie die Fähigkeit, Zielweiten einzustellen, gelernt zu werden. Während das Einstellen der Zielweite von 3m noch 200 Iterationen braucht, werden 5 m schon nach 70, 7 m nach 6 und 9 m nach 4 Iterationen eingestellt. Bei genauem Hinsehen ist aber zu erkennen, daß im ersten Versuch des neuen Blocks jeweils trotz neuer Zielweite die alte Weite produziert wird. Sollte das „Vergessen“, was in Abbildung 23f nicht zu sehen ist, auch für das Netz der Gruppe 6 gelten, kann im Test der Lernleistung und des Transfers eine stabile Produktion von 9 m erwartet werden.

#### **4.6.4 Deskriptive Auswertung**

Zur Überprüfung der drei Hypothesen werden der Lernverlauf (H 1), der Test der Lernleistung (H 2) und der Transfertest (H 3) ausgewertet. Dabei werden zuerst deskriptiv die gewonnenen Daten tabellarisch aufgeführt und graphisch dargestellt.

##### **4.6.4.1 Lernverlaufsleistung**

In Tabelle 2 sind die Mittelwerte ( $\bar{x}$ ) der Fehler zu den jeweiligen Übungszeitpunkten für die verschiedenen Gruppen und die dazugehörigen Standardabweichungen (SD) aufgetragen. Eingetragen ist der über die verschiedenen Trainingsbedingungen und Netze gemittelte Fehler bzw. dessen Standardabweichung pro Gruppe in Metern. Die Anzahl

ergibt sich aus je einem Wert pro Netz für jede der 4 Zielweiten bei 20 Netzen in 6 Gruppen und zu 4 Meßzeitpunkten. In Gruppe 1 fehlen vier Meßdaten, da bei vier Netzen 800 Iterationen durchgeführt wurden, ohne daß in einer Übungsbedingung 180 Versuche erreicht worden wären (vgl. S. 101).

Tab. 2: Beschreibende Daten des Lernverlaufs (Angaben in Metern).

	nach einem Trainingsvektor		nach 20 Trainingsvektoren		nach 100 Trainingsvektoren		nach 180 Trainingsvektoren		Anzahl
Gruppe	$\bar{x}$	SD	$\bar{x}$	SD	$\bar{x}$	SD	$\bar{x}$	SD	n
1	2,371	0,988	1,944	1,306	0,279	0,292	0,223	0,214	356
2	2,615	1,058	2,023	1,330	0,261	0,272	0,242	0,229	360
3	2,524	1,034	1,308	0,752	0,220	0,300	0,196	0,238	360
4	2,382	1,217	0,511	0,414	0,303	0,404	0,174	0,275	360
5	2,409	0,844	0,305	0,322	0,308	0,343	0,204	0,308	360
6	2,341	0,664	0,347	0,468	0,033	0,072	0,034	0,079	360
Gesamt	2,431	0,984	1,085	1,139	0,234	0,311	0,179	0,244	1916

Abbildung 24 veranschaulicht den Lernverlauf durch die Mittelwerte der Fehler über alle Kontextinterferenz-Bedingungen zu den vier Meßzeitpunkten. Dabei ist deutlich zu sehen, daß der mittlere Ausgangswert für die verschiedenen Gruppen von Netzwerken relativ gleich ist. Nach 20 Iterationen pro Versuchsbedingung haben die Netze der Gruppe 4, 5, und 6 den Fehler gegenüber der Anfangsbedingung im Mittel um ca. zwei Meter reduziert, die Netze der Gruppe 3 um einen Meter, während die Gruppen 1 und 2 die geringste Verbesserung zeigen. Dies ändert sich nach 100 Iterationen, wo die Netze der Gruppe 6 nahezu fehlerfrei die geforderte Zielweite erzeugen, derweil die Netze der anderen Gruppen einen Fehler um die 30 cm produzieren. Dieses Bild ändert sich auch nach dem vierten Meßzeitpunkt nicht.

#### 4.6.4.2 Lernleistung

Zur Erhebung der Lernleistung werden die Trainingsvektoren in das Bewegungssteuerungsnetz eingegeben, und aus dem Ausgabevektor wird die erzielte Weite errechnet. Der Abstand zwischen Soll- und Istweite wird über alle Testvektoren und über alle Netze einer Gruppe

gemittelt. Die dabei erhaltenen Daten der Fehler sind in Tabelle 3 dokumentiert, die Mittelwerte werden in Abbildung 25 graphisch dargestellt.

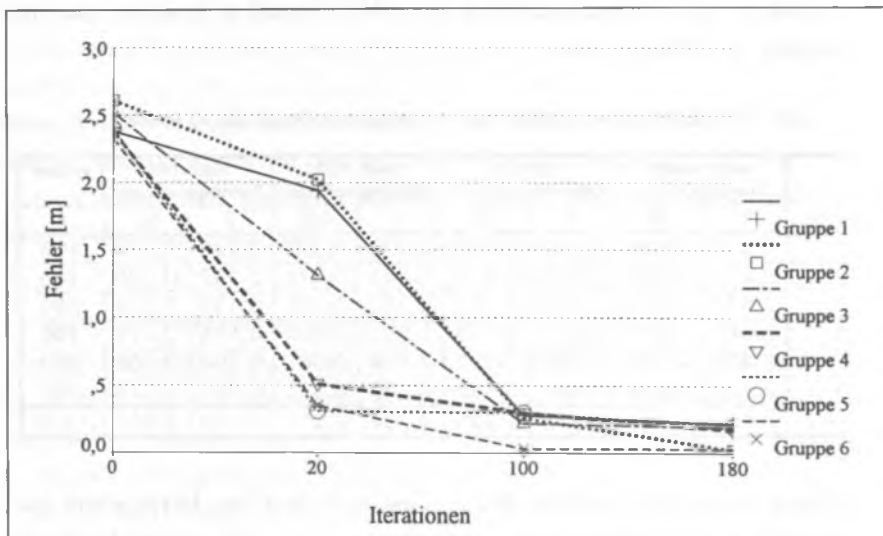


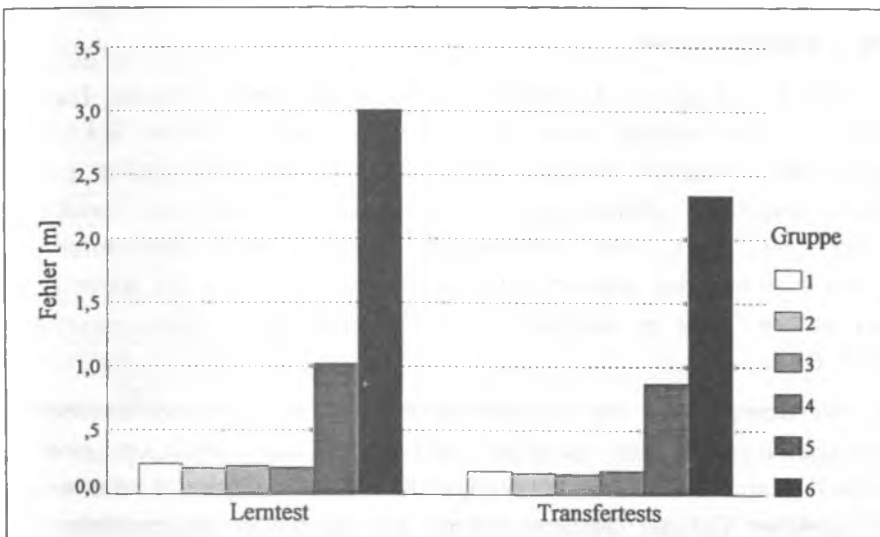
Abb. 24: Lernverläufe der Gruppen 1 - 6 (Fehler in Metern).

Auffällig ist hier die geringe Streuung der Werte der Gruppe 6, trotz des großen Fehlers. Dies bedeutet, daß das Ziel stets um einen relativ konstanten Betrag verfehlt worden ist. Darauf wird bei der Interpretation der Ergebnisse ausführlich eingegangen werden. Weiterhin wird deutlich, daß (vor allem in Gruppe 5) auch nach 800 Iterationen die Ergebnisse noch streuen, was in den Gruppen 2 bis 6 nur an der verschiedenen Initialisierung der Gewichte der einzelnen Netze liegen kann. Es ist eine nachträgliche Rechtfertigung für den Aufwand, nicht einzelne Netze, sondern Gruppen von Netzen mit Hilfe von Mittelwertstatistiken zu vergleichen.

Deutlich zu erkennen ist, daß Gruppe 5 und 6 schlechter abschneiden als die vier anderen Gruppen. Vor einer Bewertung dieser Unterschiede müssen sie aber auf ihre Signifikanz überprüft werden.

**Tab. 3: Beschreibende Daten des Fehlers im Test der Lernleistung** (alle Angaben in Metern).

Gruppe	Anzahl	$\bar{x}$	SD	Minimum	Maximum
1	20	0,2385	0,0399	0,1703	0,3438
2	20	0,2037	0,0243	0,1513	0,2352
3	20	0,2224	0,0203	0,1772	0,2650
4	20	0,2094	0,0253	0,1545	0,2525
5	20	1,0235	0,2988	0,5812	1,5485
6	20	3,0081	0,0337	2,9416	3,0754
Gesamt	120	0,8176	1,0344	0,1513	3,0754



**Abb. 25: Mittlere Fehler der Gruppen im Test der Lernleistung und der Transferleistung (in Metern).**

#### 4.6.4.3 Transfertest

Der Transfertest wird analog zum Lernleistungstest ausgeführt. Die schon in Abbildung 25 dargestellten Werte dokumentiert Tabelle 4. Auch hier ist zu erkennen, daß die Gruppen 5 und 6 schlechtere Transferergebnisse erzielen als die restlichen Gruppen.

**Tab. 4: Beschreibende Daten des Fehlers im Transfertest** (alle Angaben in Metern).

Gruppe	Anzahl	$\bar{x}$	SD	Minimum	Maximum
1	20	0,1821	0,0425	0,0829	0,2509
2	20	0,1679	0,0220	0,1354	0,2144
3	20	0,1564	0,0246	0,1050	0,2028
4	20	0,1850	0,0308	0,1415	0,2377
5	20	0,8694	0,2294	0,5287	1,2612
6	20	2,3396	0,0261	2,2880	2,3918
Gesamt	120	0,6501	0,8063	0,0829	2,3918

#### 4.6.5 Varianzanalyse

Die oben beschriebenen Unterschiede des Lernleistungs- und des Transfertests sollen mittels einer einfaktoriellen Varianzanalyse auf ihre Signifikanz überprüft werden. Sie „überprüft die Auswirkung einer p-fach gestuften, unabhängigen Variablen auf die abhängige Variable“ (Bortz 1993, 225). Die unabhängige Variable, auch Gruppenfaktor genannt, ist hier das unterschiedliche Trainingsverfahren, die abhängige Variable der Fehler im abschließenden Test. Wegen der sechs verschiedenen Gruppen ist  $p = 6$ .

Bei der Überprüfung der Lernverlaufsleistung ist eine zweifaktorielle Varianzanalyse mit dem sechsfach gestuften Gruppenfaktor und einem vierfach gestuften Meßwiederholungsfaktor sinnvoll. Dabei wird errechnet, welcher Teil der Gesamtvariation auf die beiden Haupteffekte – nämlich die Unterschiede zwischen den einzelnen Testzeitpunkten (hier: die Verbesserung) über alle Gruppen bzw. der Einfluß der Übungsbedingung über alle Testzeitpunkte – und welcher auf die Wechselwirkungen zwischen Gruppen und den verschiedenen Meßzeitpunkten – also die Unterschiede im Lernverlauf zwischen den Gruppen – zurückzuführen ist.

Die Ergebnisse der varianzanalytischen Berechnungen für Lernverlauf, Lernleistungstest und Transfertest sind in den Tabellen 5 - 7 zusammengefaßt.



Die Varianzanalyse prüft aber nur, ob die unterschiedlichen Übungsbedingungen zu statistisch signifikanten Unterschieden führen. Werden solche Signifikanzen gefunden, kann nachträglich durch den Vergleich der einzelnen Stufen des Gruppenfaktors herausgefunden werden, durch welche Gruppenunterschiede sie bedingt werden. „Durch Einzelvergleiche (oder auch Kontraste) finden wir heraus, zwischen welchen einzel-

**Tab. 5: Varianzanalyse des über die verschiedenen Zielweiten gemittelten Fehlers über die sechs Gruppen und vier Meßzeitpunkte (mit der Irrtumswahrscheinlichkeit p).**

Quelle der Variation	Quadratsumme	Freiheitsgrade	Varianz	F-Wert	p
Gruppierungsfaktor	92,71	5	18,54	31,10	<0,0005
innerhalb der Gruppen	280,24	470	0,60		
zwischen den Netzen	372,95	475			
Meßwiederholung	1584,45	3	528,15	1241,55	<0,0005
Gruppe × Meßwiederholung	182,03	15	12,14	28,53	<0,0005
Meßwiederholung × Netze	599,80	1410	0,43		
innerhalb der Netze	2366,28	1428			
Total	2739,23	1903			

**Tab. 6: Varianzanalyse des über die verschiedenen Zielweiten gemittelten Fehlers bei dem Lerntest über die sechs Gruppen.**

Quelle der Variation	Freiheitsgrade	Quadratsumme	Varianz	F-Wert	p
Gruppenfaktor	5	125,54	25,11	1608,37	< 0,00005
Fehler	114	1,78	0,0156		
Total	119	127,32			

**Tab. 7: Varianzanalyse des über die verschiedenen Zielweiten gemittelten Fehlers bei dem Transfertest über die sechs Gruppen.**

Quelle der Variation	Freiheitsgrade	Quadratsumme	Varianz	F-Wert	p
Gruppenfaktor	5	76,28	15,26	1601,32	<0,00005
Fehler	114	1,09	0,01		
Total	119	77,37			

nen Treatmentstufen signifikante Unterschiede bestehen“ (Bortz 1993, 240). Aus der Reihe der möglichen Methoden wird der Scheffé-Test gewählt, die Ergebnisse werden für die einfaktoriellen Varianzanalysen in den Tabellen 8a und b dargestellt. Dabei werden die Gruppen der Größe des Mittelwerts nach geordnet, ein Stern in der Tafel bedeutet, daß zwischen den beiden Gruppen ein signifikanter Unterschied besteht. Die wenig anschauliche Tabelle für die 276 möglichen Einzelvergleiche der Interaktionen (24 Zellenmittelwerte, jeder wird mit jedem verglichen) bleibt der Leserin und dem Leser erspart.

Tab. 8: Ergebnisse des Scheffé-Tests für

a) den Test der Lernleistung und

Gruppe	$\bar{x}$	2	4	3	1	5	6
2	0,204						
4	0,209						
3	0,222						
1	0,238						
5	1,023	*	*	*	*		
6	3,008	*	*	*	*	*	

b) den Transfertest

Gruppe	$\bar{x}$	3	2	1	4	5	6
3	0,156						
2	0,168						
1	0,182						
4	0,185						
5	0,869	*	*	*	*		
6	2,340	*	*	*	*	*	

#### 4.6.6 Interpretation

In dieser Simulation werden die aus dem Kontextinterferenz-Effekt abgeleiteten drei Hypothesen bestätigt, wie auch schon bei Masson (1990) und Horak (1992) an einzelnen Netzen veranschaulicht. Die in Abbildung 25 dargestellten Unterschiede erweisen sich als hochsignifikant. Höhere Kontextinterferenz führt also auch bei dem Lernen im Netz zu besseren Ergebnissen im Test des Lernergebnisses und des Transfers. Dabei kann mit an Sicherheit grenzender Wahrscheinlichkeit ausgeschlossen werden, daß es sich um einen zufälligen Effekt handelt. Die Signifikanz der Interaktion sagt aus, daß sich die einzelnen Gruppen hinsichtlich ihres Lernverlaufs überzufällig unterscheiden.

Die a-posteriori-Einzelvergleiche bestätigen dieses Ergebnis. Bei den einfaktoriellen Designs können leicht jeweils Untergruppen gebildet

werden, innerhalb derer sich der höchste und der niedrigste Gruppenmittelwert nicht signifikant unterscheiden. Im Lernleistungs- und Transfertest werden drei Untergruppen diagnostiziert, die Gruppen 1 - 4, Gruppe 5 und Gruppe 6. Während zwischen den Gruppen 1 - 4 das unterschiedliche Training zu keinen überzufälligen Unterschieden bezüglich der Testleistung führt, ist die in Blöcken mit 10 Wiederholungen übende Netzgruppe 5 signifikant schlechter als diese, jedoch wiederum signifikant besser als Gruppe 6.

Bei den Einzelvergleichen der Interaktion zwischen dem Gruppen- und dem Meßwiederholungsfaktor wird die Signifikanz der Unterschiede zwischen den Gruppen bezüglich der Veränderung zwischen den Meßzeitpunkten ermittelt. Zwischen dem Anfangstest und dem zweiten Meßzeitpunkt nach 20 Iterationen unterscheiden sich drei homogene Untergruppen voneinander. Die erste wird von den Gruppen 1 und 2, die zweite von der Gruppe 3 und die dritte von den Gruppen 4, 5 und 6 gebildet. In Zusammenschau mit der Graphik kann interpretiert werden, daß die Netze der Gruppen 4, 5 und 6 signifikant schneller lernen als die Netze der Gruppe 3, diese wiederum lernen signifikant schneller als Netze der Gruppen 1 und 2. Die Unterschiede innerhalb der Untergruppen sind statistisch jeweils nicht bedeutsam. Ein ähnliches, aber umgekehrtes Bild ergibt sich zwischen Meßzeitpunkt zwei und drei. Hier bilden sich vier Untergruppen. Am meisten verbessern sich Netze der Gruppen 1 und 2, danach die Gruppe 3, gefolgt von den Gruppen 4 und 6. Netze der Gruppe 5 weisen im Schnitt keine Verbesserung auf. Zwischen dem dritten und dem vierten Meßzeitpunkt verbessert sich nur noch die Gruppe 4 signifikant, in der Lernrate unterscheidet sie sich überzufällig von Gruppe 6. Zur Bestätigung der Hypothese 1 reicht die Betrachtung der Interaktion zwischen dem ersten und dem zweiten Meßzeitpunkt. Es bestätigt sich in der statistischen Analyse, daß Gruppen mit höherer Kontextinterferenz signifikant langsamer lernen als Gruppen mit niedriger Kontextinterferenz.

Somit kann festgestellt werden, daß alle aus dem beim menschlichen Lernen beobachteten Kontextinterferenz-Effekt abgeleiteten Hypothesen beim Lernen eines neuronalen Netzwerks bestätigt werden können.

Trotzdem bleiben einige Merkwürdigkeiten, die bei einer Interpretation dieses Befunds berücksichtigt werden sollten: Das bei Betrachtung der Lernverläufe beobachtete „Vergessen“ und die kleine Streuung der Gruppe 6 im Lern- und Transfertest. Der Grund für letzteres wird deutlich, wenn man die zuvor gemittelten Fehler nach den verschiedenen Zielweiten aufschlüsselt. Die dann erhaltenen Werte aus Tabelle 9 werden in Abbildung 26 verdeutlicht.

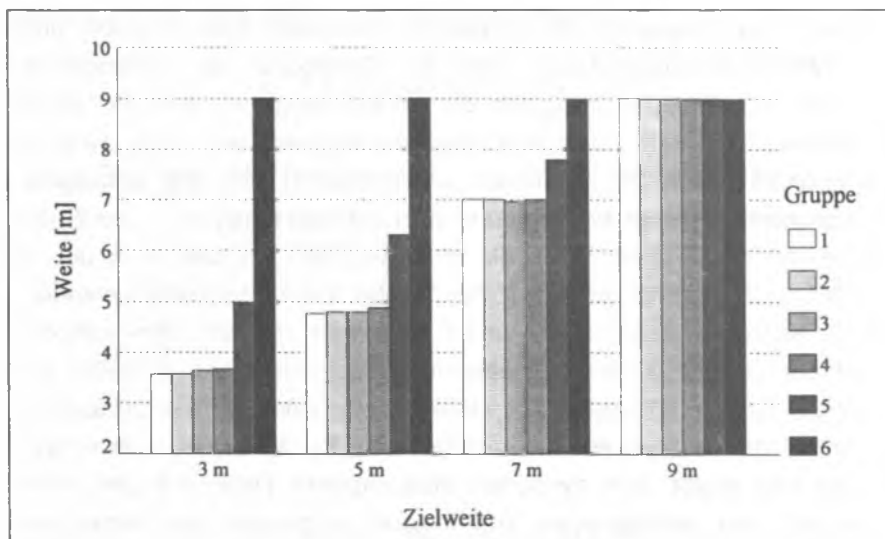


Abb. 26: Die erreichten Weiten (in Metern) aufgeschlüsselt nach Gruppen und Zielweiten.

Hier wird deutlich, daß Gruppe 6 ein so schlechtes Lernergebnis zeigt, weil im Mittel (und mit sehr geringer Streuung) für jede Zielweite 9 m, das Ziel des letzten Lernblocks, produziert wird. Die Lernblöcke davor sind anscheinend komplett „vergessen“ worden. Dieses „Vergessen“ muß auf der Ebene der Netzwerke so interpretiert werden, daß sich die Gewichte neu auf die aktuelle Zielweite einstellen. Im vorliegenden Fall ist es so, daß bei mehr als fünf Wiederholungen eines Trainingsvektors in einem Block sich die Gewichte neu organisieren und die alten Verbindungen zwischen den Neuronen aufgehoben werden. Dieses Verhalten

Tab. 9: Erreichte Weiten und die dazugehörigen Standardabweichungen (beide in Metern), aufgeschlüsselt nach den Gruppen und den zu erreichenden Zielweiten.

Ziel	Gruppe	1	2	3	4	5	6
3 m	$\bar{x}$	3,575	3,583	3,642	3,673	4,987	9,016
	SD	0,100	0,089	0,066	0,106	0,532	0,068
5 m	$\bar{x}$	4,772	4,801	4,805	4,884	6,309	9,010
	SD	0,092	0,013	0,008	0,030	0,447	0,045
7 m	$\bar{x}$	7,014	7,009	6,966	7,008	7,798	9,005
	SD	0,122	0,032	0,055	0,062	0,226	0,022
9 m	$\bar{x}$	9,0070	8,997	9,001	9,000	9,000	9,000
	SD	0,086	0,121	0,003	0,000	0,000	0,000

muß auch bei Masson und Horak zur Bestätigung des Kontextinterferenz-Effekt geführt haben, was von Masson allerdings verschwiegen wird. Horak merkt aber an:

On transfer, the network had a predominant local fit for the last distance and gave its prediction from this last function (1992, 63).

Massons zweiter Test der Lernleistung (vgl. Abb. 16, S. 86) muß vor diesem Hintergrund als eine unbedachte Anwendung eines konnektionistischen Modells gekennzeichnet werden. Bei den meisten Experimenten zum Kontextinterferenz-Effekt für menschliches Lernen wird (mindestens) ein Behaltenstest in einem gewissen Zeitabstand nach dem Training durchgeführt. Dies ist für Simulationen auf neuronalen Netzwerken aber nicht sinnvoll, die Lernleistung eines Computers ändert sich nicht, wenn man ihn eine Stunde oder gar zwei Wochen „ruhen“ läßt. Die Verbesserungen, die Masson in seinem „Behaltenstest“ feststellt, beruhen lediglich darauf, daß er im ersten Test nicht die Gewichte konstant hält und so noch ein weiterer Lerndurchgang mit hoher Kontextinterferenz stattfindet.

Zusammenfassend kann festgestellt werden, daß der für das menschliche Lernen gefundene Kontextinterferenz-Effekt auch für das Lernen neuronaler Netzwerke gilt. Ein Rückschluß auf die gleichen Lernvorgänge erscheint aber zweifelhaft. So ist völlig selbstverständlich, daß

ein Mensch, der noch so viele Wiederholungen eines Basketballwurfs auf kurze Distanz geübt hat, schon beim ersten Versuch auf einen weiter entfernten Korb nicht die gleiche kurze Distanz der Würfe zuvor wiederholt, sondern einfach weiter werfen wird. Allerdings wird durch die obige Analyse des Lernmechanismus' neuronaler Netze eine Analogie zu menschlichen Lernverfahren nicht völlig widerlegt: Irgendwann muß schließlich auch ein Mensch gelernt haben, daß man kräftiger werfen muß, um einen weiter entfernten Gegenstand zu treffen. Eine jahrelange Vorerfahrung, über die alle Menschen, die das erste Mal einen Basketball zur Hand nehmen, schon verfügen, hat ein neuronales Netz eben noch nicht erworben.

#### 4.7 Freiheitsgrade

In der gestellten Lernaufgabe sollen die zwei Parameter Abfluggeschwindigkeit und Abflugwinkel durch eine Zielvariable, die gewünschte Weite, bestimmt werden. Damit besitzt dieses Gleichungssystem einen unbestimmbaren Freiheitsgrad. Alle Winkel-Geschwindigkeits-Paare, die für eine gegebene Weite Gleichung 18 erfüllen, lösen die Lernaufgabe. In den oben vorgestellten Simulationen zum Kontextinterferenz-Effekt wird es dabei der Netzdynamik überlassen, welche Parameterkombination sich einstellt. Lediglich die Normierungsfunktion legt die Randbedingungen fest. Eine Erregung von  $-\infty$  entspräche der Ausgabe des Neurons von -1 (s. Abb. 8 auf S. 44), die durch die gewählte Entnormierungsfunktion (die Erregung von 0,1 entspricht  $40^\circ$  bzw. 7 m/s) in  $37,5$  Grad bzw. in 6,5 m/s (abhängig von dem Neuron, von dem die Ausgabe produziert wird) zurücktransformiert wird. Dementsprechend ist die Obergrenze  $62,5$  Grad und 12,5 m/s.

Zunächst soll betrachtet werden, wie sich die Netze ohne eine Beeinflussung der Parameterwahl verhalten. Dazu sollen die Abbildungen 27a-h dienen.

Die Balken stellen die Anzahl derjenigen Netze dar, deren Parameter im Bereich des auf der Abszisse angegebenen Wertes liegt. Dabei ist bei

den den Geschwindigkeitsparameter betreffenden Diagrammen der Bereich um den angegebenen Wert  $\pm 0,125$  m/s, für den Winkelparameter  $\pm 0,5^\circ$ .

Macht man sich die Mühe, die Diagramme ein wenig genauer zu betrachten, so bemerkt man, daß die Winkel mehr oder weniger über

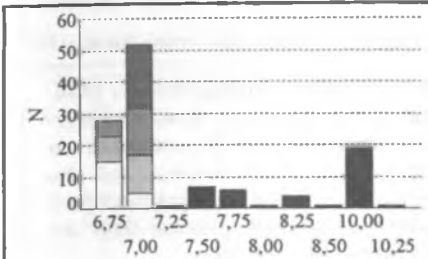


Abb. 27a:  $v_0$  für die Zielweite 3 m

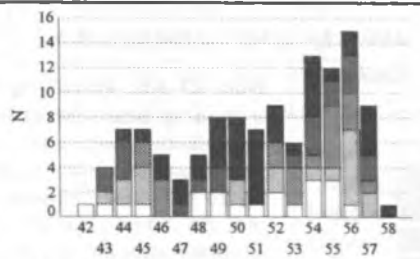


Abb. 27b:  $\alpha$  für die Zielweite 3 m

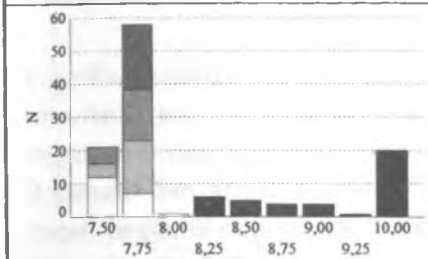


Abb. 27c:  $v_0$  für die Zielweite 5 m

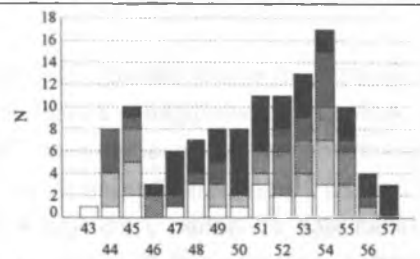


Abb. 27d:  $\alpha$  für die Zielweite 5 m

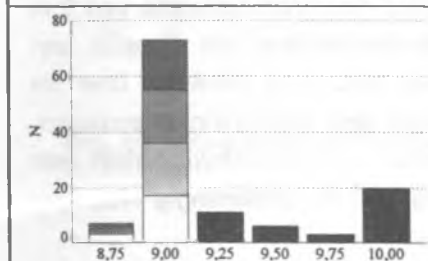


Abb. 27e:  $v_0$  für die Zielweite 7 m

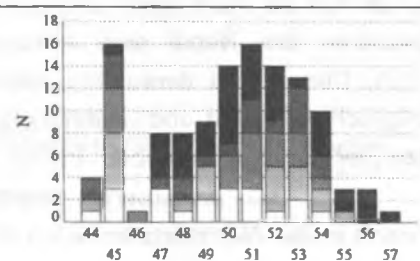
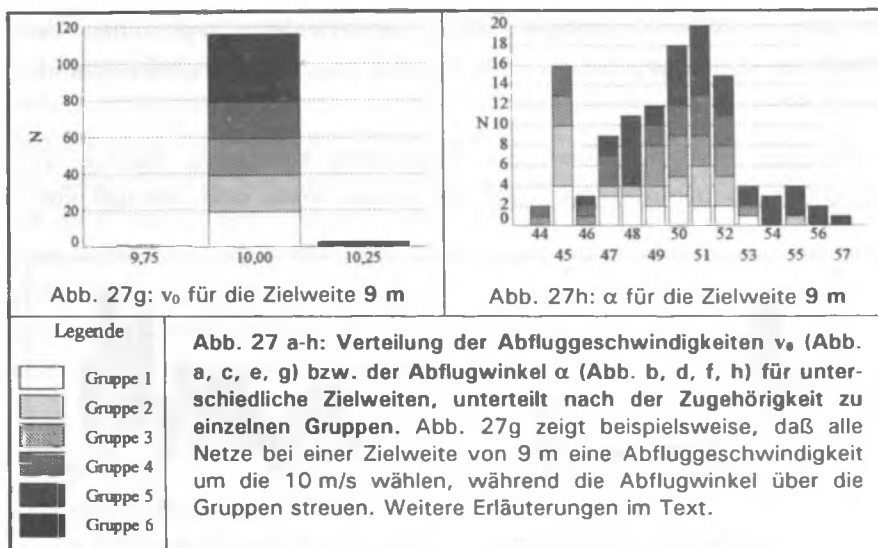


Abb. 27f:  $\alpha$  für die Zielweite 7 m



alle Gruppen in einem Bereich zwischen  $42^\circ$  und  $58^\circ$  streuen, während die Abfluggeschwindigkeit zwischen den Gruppen je nach Zielweite unterschiedliche Werte annehmen. Bei einer Zielweite von 3 m liegen 19 der 20 Netze aus Gruppe 6 bei 10 m/s, die Netze der Gruppe 5 verteilen sich zwischen 7,25 und 8,5 m/s, alle übrigen Gruppen liegen ungefähr zwischen 6,75 und 7 m/s (auf die Balkenbreite von  $\pm 0,125$  m/s sei nicht mehr eingegangen). Bei einer Zielweite von 9 m produzieren alle Netze eine Abfluggeschwindigkeit im Bereich von 10 m/s. Dies deutet darauf hin, daß die Netze die Zielweite über die Wurfgeschwindigkeit und unabhängig von dem Abflugwinkel erzeugen. Netze der Gruppe 6 geben immer eine Abfluggeschwindigkeit von 10 m/s aus, sie produzieren allerdings auch unabhängig von der Zielweite in den Nachtests konstant 9 m.

Netze der Gruppen 1 bis 4 passen ihre Abfluggeschwindigkeit an die Zielweite an, von ungefähr 7 m/s bei einer Zielweite von 3 m über 7,75 m/s und 9 m/s (für 5 und 7 m) bis 10 m/s bei einer Zielweite von 9 m. Dieses Verhalten ergibt sich aus der gewählten Aufgabe, es kann in Abbildung 28 verdeutlicht werden. Angenommen, das Netz produ-



ziert als Ausgabe eine Abfluggeschwindigkeit von 6,5 m/s und einen Abflugwinkel von  $42^\circ$ . Um die Zielweite von 3 m zu erreichen, müßte der Winkel (bei konstanter Geschwindigkeit) um ca.  $7^\circ$  auf  $49^\circ$  korrigiert werden, die Abfluggeschwindigkeit (bei konstantem Winkel) jedoch nur um ca. 0,3 m/s. Bei einer größeren Zielweite als 3 m wäre das Ziel über eine alleinige Winkeländerung überhaupt nicht zu erreichen. Für das Verhalten eines Netzes, das mit dem *backpropagation*-Algorithmus lernt, bedeutet dies, daß in der Fehlerlandschaft die Steigung auf der Geschwindigkeitsachse betragsmäßig größer ist als auf der Winkelachse. Das heißt in der bildlichen Interpretation, daß die „Fallinie“, auf

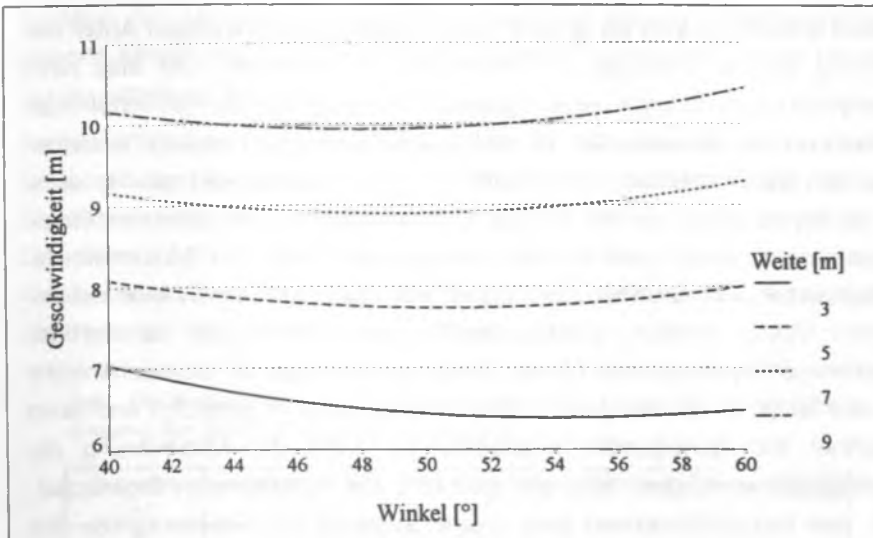


Abb. 28: Abflugwinkel und Geschwindigkeiten (in Metern pro Sekunde) für verschiedene Zielweiten (dies entspricht den Höhenlinien der jeweiligen Weiten in Abbildung 18 auf S. 93).

der die Kugel talabwärts rollt (vgl. S. 47), mehr in Richtung Geschwindigkeits- und weniger in Richtung Winkelachse weist. Daher lernen die Netze verstärkt, die Zielweite über eine Änderung der Abfluggeschwindigkeit anzusteuern.

Im folgenden soll der Versuch unternommen werden, auf die Parame-

terwahl Einfluß zu nehmen, das „Problem des Freiheitsgrads“ zu lösen. Dazu soll angenommen werden, daß es sinnvoll sei, die Abfluggeschwindigkeit zu minimieren, das einzige Kriterium, das passend und gleichzeitig nicht zu schwer zu simulieren ist. Prinzipielles Vorgehen ist es, aus der Optimierungs- oder Randbedingung Fehlerterme zu errechnen, die dann im Netz verrechnet werden. Anders als Jordan, der die aus mehreren Randbedingungen erzeugten Fehlerterme an verschiedenen Stellen dem Netz bekannt macht (Jordan 1989, 810), soll hier eine Randbedingung ausreichen, um den prinzipiellen Lösungsansatz für das Problem der Freiheitsgrade in konnektionistischen Modellen zu demonstrieren.

Dabei erweist es sich als größte Schwierigkeit, den jeweiligen Anteil des Fehlers an der Erfüllung der Bedingung zu ermitteln. Um dies nicht analytisch zu tun, kann die Ausgabeschicht des Bewegungsvorstellungsnetzes um ein Neuron, das die Abfluggeschwindigkeit codiert, erweitert werden. Die Abbildung der Eingabe- auf die Ausgabevektoren ist dabei leider etwas trivial, da die Abfluggeschwindigkeit auch schon als Eingabeparameter dient, weil in dem vorliegenden Netz der Muskelaktivierungsvektor aus Gründen der Vereinfachung durch einen biomechanischen Vektor ersetzt wurde. Besäße man jedoch ein vernünftiges anatomisch-dynamisches Modell (und das vorliegende biomechanische Modell ist ja als starke Vereinfachung eines solchen gedacht) und einen Rechner mit genügender Kapazität, so wäre die Minimierung der Abfluggeschwindigkeit sinnvoll, benutzte die nichtlinearen Eigenschaften von Netzsimulationen und würde zugleich die Freiheitsgrade des anatomisch-dynamischen Modells stark einschränken.

Das neue Netzwerk zur Lösung des Problem der Freiheitsgrade entspricht also dem in Abbildung 21 (S. 96) dargestellten, hat aber bei der Ausgabe des Bewegungsvorstellungsnetzes zusätzlich zu dem Neuron N1, das die Wurfweite abbildet, ein weiteres Neuron N2, das die Abfluggeschwindigkeit lernt. Nach dem Training des Bewegungsvorstellungsnetzes, wiederum 2000 Iterationen mit je 500 zufällig ausgewählten Trainingsvektoren, kann das Erlernen der Bewegungssteuerung simuliert werden. Während N1 weiterhin den Fehler zwischen tatsächli-

cher und geforderter Weite zurückmeldet („zu weit“ bzw. „zu kurz“), liefert N2 stets eine zu hohe Abfluggeschwindigkeit („zu schnell“) zurück. Dies wird dadurch erreicht, daß als korrektes Ergebnis immer eine Geschwindigkeit gewählt wird, die etwas unter der erreichten liegt. Dabei muß darauf geachtet werden, daß der Fehler der Randbedingung im Vergleich zu dem Fehler der eigentlichen Aufgabe klein bleibt, da nicht die Randbedingung gelernt werden soll (vgl. Jordan 1992). Eine gewisse Abfluggeschwindigkeit ist schließlich erforderlich, um den Korb zu treffen. Trotzdem würde so die Zielfunktion, d. h. korrekte Abflugwinkel und -geschwindigkeit, nicht genau gelernt, da das Minimum der rückgemeldeten Fehlerfunktion nicht das Minimum der zu erlernenden Funktion ist. Um dies zu erreichen, wird der als die Randbedingung addierte Fehler über den Lernverlauf verkleinert, bis er schließlich bei der letzten Iteration Null ist.

An vier Netzwerken wird versucht, durch Modellierung von äußerer Rückmeldung („zu schnell“) das Problem des einen Freiheitsgrads der Simulationsaufgabe zu lösen. Jedes Netz trainiert eine unterschiedliche Zielweite: 3, 5, 7 und 9 m. Nach 2000 Iterationen wird getestet, welche Winkel-Geschwindigkeits-Kombination sich eingestellt hat. Die Lösungen der Netze wird in Tab. 10 dargestellt.

**Tab. 10: Abflugwinkel und -geschwindigkeit für Netze mit der Randbedingung der Geschwindigkeitsminimierung.**

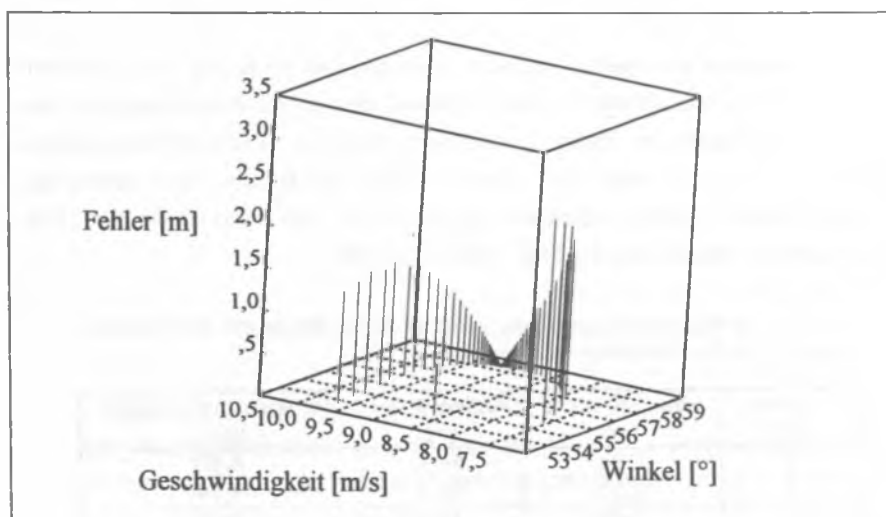
Zielweite [m]	erreichte Weite [m]	Abflugwinkel [°]	Abfluggeschwindigkeit [m/s]
3	3,04	59,5	6,50
5	5,00	50,4	7,74
7	7,00	49,1	8,90
9	9,00	47,1	9,93

Vergleicht man die produzierten Flugparameter mit Abbildung 28, so ist zu erkennen, daß für die Zielweiten 5, 7 und 9 m das Geschwindigkeitsminimum in guter Näherung gefunden wird. Lediglich für eine Korbentfernung von 3 m kann dies nicht behauptet werden, was vermutlich daran liegt, daß hier in einem großen Bereich eine Winkeländerung kaum eine Veränderungen der Geschwindigkeit mit sich bringt.

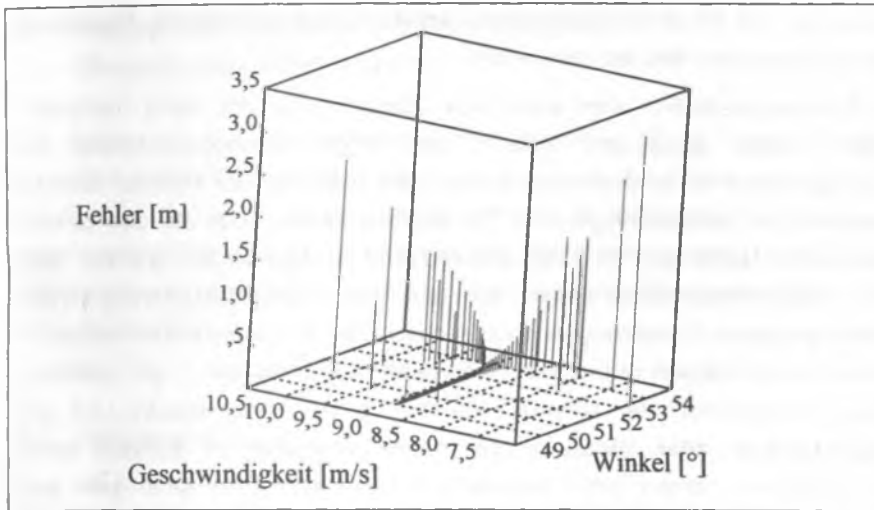
Der Unterschied zwischen den Netzen mit und ohne Einbeziehung des Optimierungskriteriums soll hier nur anschaulich deutlich gemacht werden. Abbildung 29a und 29b stellen vergleichend den Lernverlauf eines Netzes ohne und mit Berücksichtigung des Problems der Freiheitsgrade dar.

Im Vergleich ist deutlich zu erkennen, daß das Netz ohne Geschwindigkeitsminimierung wesentlich schneller lernt, das Netz mit diesem Optimierungskriterium zwar langsamer lernt, aber einen Winkel im optimalen Bereich findet. Die Langsamkeit des Lernens verursacht die durch die Randbedingung verfälschte Fehlerfunktion.

Nebenbei wird hier der Einfluß der Lernkonstanten  $\gamma$  anschaulich dargestellt. Zum Erlernen des Bewegungssteuerungsnetzes mit der Randbe



**Abb. 29a:** Lernverlauf eines Netzes ohne Berücksichtigung der Randbedingung. Die Zielweite von 7 m erreicht das Netz mit einem Abflugwinkel von 58,4° und einer Abfluggeschwindigkeit von 9,2 m/s.



**Abb. 29b:** Lernverlauf eines Netzes mit Berücksichtigung der Randbedingung. Die Zielweite von 7 m erreicht das Netz mit einem Abflugwinkel von  $49,1^\circ$  und einer Abfluggeschwindigkeit von 8,9 m/s.

dingung wurde  $\gamma = 20$  gewählt. Damit ist sie offensichtlich so groß, daß die „Kugel“ zwischen zwei „Hängen“ der Fehlerlandschaft hin und her hüpfte, bis sie ins „Tal“ findet.

#### 4.8 Einordnung und Bewertung

Die hier vorgestellte Simulation von Bewegungsdaten in einem neuronalen Netzwerk benutzt die Architektur von Jordans (1989) Netz. Es gehört in die Gruppe der Netze mit überwachtem Lernen. Die Überwachung geschieht durch eine in einem Netz repräsentierte Vorstellung der Umgebung; das Lernen von Bewegungen wird so zum Lernen der Inversen dieser Bewegungsvorstellung. Jordans Architektur wird vereinfacht um das Konzept der rekurrenten Verbindungen, damit entfällt die Möglichkeit, sequentielle Bewegungen oder gar deren Dynamik zu simulieren. Folgerichtig werden in dem Modell der Bewegung auch keine

dynamischen Muskelaktivierungen, sondern biomechanische Parameter als Ausgabe des Netzes verwendet.

In Simulationsläufen wird empirisch überprüft, ob der beim menschlichen Lernen gefundene Effekt, daß hohe Kontextinterferenz im Übungsverlauf zu schlechteren Leistungen während der Übung, aber zu besseren im Lernleistungs- und Transfertest führt, auch für das Lernen neuronaler Netze gilt. Es kann überzeugend nachgewiesen werden, daß die zusammengefaßten Daten den aus dem Kontextinterferenz-Effekt vorhergesagten Erwartungen entsprechen. Die dabei teilweise beobachteten Lernstrategien und -gesetzmäßigkeiten des Netzes – ein vollständiges „Vergessen“ der Übungsversuche vergangender Blöcke und ein Reproduzieren alter Versuche bei neuen Aufgaben – können beim menschlichen Lernen nicht angenommen werden. Dies kann sehr gut damit erklärt werden, daß die trainierten neuronalen Netze über keinerlei „Bewegungserfahrung“ verfügen, bevor sie die Übungen absolvieren.

Innerhalb des Modells wird durch Verwendung externer Randbedingungen das Problem der Freiheitsgrade gelöst. Diese Lösung gelingt nicht besonders überzeugend, da das Modell schon so weit vereinfacht ist, daß nur noch ein Freiheitsgrad bestimmt werden muß, die Lösung nur anschaulich dokumentiert, aber nicht statistisch nachgewiesen wird und nur in drei von vier Simulationsfällen gelingt.

Der Wert der hier erstellten Simulation liegt also hauptsächlich auf der Aufdeckung der Mechanismen, die beim Lernen in neuronalen Netzwerken zu Kontextinterferenz-Effekten führen und der Veranschaulichung einer prinzipiellen Vorgehensweise zur Lösung des Problems der Freiheitsgrade in einem konnektionistischen Modell.

## 5 Perspektiven konnektionistischer Bewegungsforschung

Die Bedeutung des konnektionistischen Ansatzes für die sportwissenschaftliche Bewegungsforschung ist schwierig einzuschätzen. Trotzdem sei hier der Versuch gewagt, das gegenwärtige Meinungsbild wiederzugeben und Stärken und Schwächen des Konnektionismus, seine Auswirkungen auf die Erforschung sportpraktischer Fragestellungen und seine Perspektive zu beurteilen.

### 5.1 Bedeutung

Zunächst muß hier auf die Bewertung des Konnektionismus innerhalb der Psychologie eingegangen werden, denn ohne die Unterstützung der Mutterwissenschaft wird sich ein Modell nicht in der Sportwissenschaft halten können. Diese Bewertung ist stark abhängig von der Ebene, für die diesem Ansatz Erklärungswert zugesprochen wird. Es wird allgemein anerkannt, daß die Erläuterung des algorithmischen Levels im Sinne Marrs (1982, vgl. Kapitel 1) die Aufgabe der Psychologie sei (z.B. Rumelhart & McClelland 1986, 123), daß aber Konzepte rein auf der Implementationsebene keine gültigen Aussagen über psychische Phänomene liefern können. Über die Zuordnung des Konnektionismus wurde noch in jüngerer Vergangenheit erbittert gestritten:

- Rumelhart & McClelland (1986) vertreten die Meinung, daß Marrs verschiedene Ebenen stark miteinander zusammenhingen und nicht getrennt zu analysieren seien (S. 123). Der Konnektionismus liege also in gewisser Hinsicht zwischen den Ebenen. Obwohl sich der Konnektionismus mit der *microstructure of cognition* (Untertitel von Rumelhart, McClelland u.a. bzw. McClelland, Rumelhart u.a. 1986) beschäftige, biete er Beschreibungs- und Erklärungswert für Vorgänge auf der algorithmischen Ebene.
- Smolensky (1988) ist der Auffassung, daß „the level of PTC [the proper treatment of connectionism] is more explicitly related to the

level of the symbolic paradigm [i.e. the algorithmic level] than it is to the neural [implementational] level" (S. 3). Auch Heuer (1992, 758) bemerkt: „Connectionist networks, however, belong to the algorithmic level of explanation.“

- Die Sicht derjenigen, die den Wert konnektionistischer Modellbildung skeptisch einschätzen, faßt Keller (1990, 317) zusammen:

Zusammenfassend können wir festhalten, daß eine Kognitionstheorie, die keine Möglichkeit der Abbildung strukturierter Repräsentationen und struktursensitiver Prozesse erlaubt, nicht angemessen sein kann. Konnektionistische Modelle kommen damit nicht als psychologische Modelle der Kognition infrage. Sie stellen eher potentielle Theorien der Implementation einer Kognitionstheorie dar.

Dabei läßt sich beobachten, daß Psychologen, die sich hauptsächlich mit Themen wie Sprachverarbeitung und mathematischer Problemlösung beschäftigen, zu der Auffassung von Keller neigen, während Forscher im Bereich menschlicher Wahrnehmung eher Smolensky zustimmen.

Dies deutet eine Entwicklung an, die sich durch alle Bewertungen der Aussagekraft konnektionistischer Modellbildung hindurch beobachten läßt. Sie wurde schon von Smolensky (1988) angesprochen. Danach kann formal zwischen zwei „Prozessoren“ unterschieden werden, mit denen menschliche Kognition modelliert werden muß: Der „bewußte Regelinterpretierer“ (*conscious rule interpreter*), der kulturelles Wissen prozessiert und bewußtes Auswerten von Regeln leistet und der „intuitive Prozessor“ (*intuitive processor*), der individuelles Wissen, Intuition und Fertigkeiten (*skill*) verarbeitet.

Die Frage, welcher der beiden Prozessoren nun für die Verarbeitung motorisch-kognitiver Prozesse zuständig sei, kann sicher nicht eindeutig zugunsten des einen oder des anderen entschieden werden, allerdings neigt sich die Waagschale deutlich auf die Seite des intuitiven Prozessors. Automatisierte Bewegungen, also Bewegungen, die nicht bewußtseinspflichtig sind, werden folglich auch nicht durch einen bewußten Regelinterpretierer gesteuert. Auch für den Lernvorgang spielt das Bewußtsein überraschenderweise nicht unbedingt eine dominierende



Rolle. So kann nur schwerlich behauptet werden, daß Kinder, wenn sie laufen oder sprechen lernen, bewußten Regeln folgen. Aber auch bei dem Erlernen komplexer spieltaktischer Regeln stellen Roth und Raab (1994) die Frage: Geschieht es „mühsam, konzentriert, intentional oder mühelos, nebensächlich, inzidentell?“

In ihrer Polemik gegen den Konnektionismus geben Fodor und Pylyshyn (1988, 52) zu, daß das Fehlen eines Fortschritts bei der Behandlung von nonverbalen oder intuitiven Prozessen des klassischen informationsverarbeitenden Ansatzes eine der Verführungen (*lures*) des Konnektionismus sei (womit sie „implizit“ zugeben, daß auch sie davon ausgehen, solche Prozesse seien durch konnektionistische Modelle besser abbildbar). Sie führen weiter aus:

The one thing that Classical theorists do agree about is that it *can't* be that *all* behavioral regularities are determined by explicit rules; at least some of the causal determinants of compliant behavior *must* be *implicit*. (...) The basic point is this: not all the functions of a Classical computer can be encoded in the form of an explicit program; some of them must be wired in. (...) In such cases, Classical machines can be *rule implicit* with respect to their programs (...). So, then, you can't attack Classical theories of cognitive architecture by showing that a cognitive process is rule-implicit; Classical architecture *permits* rule-explicit processes but does *not* require them (Fodor & Pylyshyn 1988, 60f).

Da sie davon ausgehen, explizite Regelverarbeitung sei für konnektionistische Modelle unmöglich, kommen sie zu dem Schluß, daß nur das klassische Paradigma die adäquate Herangehensweise sei.

Unbeeindruckt davon setzt sich aber zunehmend die Erkenntnis durch, daß intuitives, nonverbales Wissen und Bewegungssteuerung *die* Domäne der Modellbildung durch neuronale Netze ist. Dies läßt sich mit vielen Zitaten belegen, z.B.:

Thus, connectionism certainly has the potential to foster the study of relationships between perception and action (Neumann & Prinz 1990, 16).

Therefore, it is not surprising that PDP models have been very successful when they have been applied to (...) perceptual and motor processes (Parks u.a. 1991, 210).

It is regularly observed that connectionist networks are good for explaining "automatic" processes such as perception and motor control (Rutkowska 1992, 26).

Durch die Übernahme des konnektionistischen Ansatzes rücken neue Fragen in den Mittelpunkt wissenschaftlichen Erkenntnisinteresses. Menschliches Bewegungsverhalten spielt in der Kognitiven Psychologie bisher eine sehr periphere Rolle (vgl. Neumann 1993, 61; Holzkamp 1993, 271). Dies ist auch nicht besonders verwunderlich, denn selbst „niedere“ Tiere vollbringen erstaunliche Bewegungsleistungen, obwohl ihnen keine kognitiven Fähigkeiten zugeschrieben werden, woraus folgt, daß Sich-Bewegen nicht notwendig eine kognitive Leistung ist (zur Beziehung von Kognition und Motorik vgl. Munzert 1992). Eine „konnektionistische Wende“ wird die Frage nach der Bewegung wieder mehr in den Mittelpunkt stellen.

In der theoretischen Psychologie treten neuronale Modelle an die Stelle der Informationsverarbeitungs-Modelle. (...) Und parallel dazu interessiert man sich, was die Forschungsthematik angeht, zunehmend für periphere und insbesondere motorische Prozesse, d.h. für Vorgänge am realen biologischen Lebewesen mit seiner spezifischen Ausstattung mit Effektoren, deren spezifischer Biomechanik und der Notwendigkeit, ihre Steuerung an spezifische Bewegungsanforderungen anzupassen (Neumann 1993, 67).

Wenn sich also das neuronale Netzwerk als adäquates (wenn auch nicht ausschließliches) Modell für psychische Funktionen durchsetzt, wird auch die Motorikforschung im Sog einer konnektionistischen Psychologie einen gewaltigen Schub nach vorn erfahren. Dies wird auch in der sportwissenschaftlichen Bewegungswissenschaft dem Modellieren mit neuronalen Netzen eine wachsende Bedeutung zukommen lassen.

## **5.2 Stärken**

In Kapitel 1.5 auf Seite 33 wurden vier Forderungen an ein Modell gestellt, das die aktuellen Fragen der Bewegungswissenschaft beantworten will. Man könnte fast vermuten, daß sie schon in Kenntnis der Eigenschaften des Konnektionismus aufgestellt wurden, so sehr er-

scheinen sie ihm auf den Leib geschnitten. Wie in dieser Arbeit deutlich wurde,

- ist die Motorik eines der vielversprechendsten Anwendungsgebiete der Modellbildung mit neuronalen Netzen
- ist Lernen die zentrale Kategorie in diesem Modell
- werden anatomische und neurophysiologische Überlegungen in die Modellbildung mit einbezogen und
- werden durch Simulation auf Computern konkrete Aussagen zur Bewegungsausführung möglich.

Die wichtigste Bedeutung für die Bewegungswissenschaft hat aber die Möglichkeit der konnektionistischen Herangehensweise, bestehende Konzepte aus allen Ebenen der Theoriebildung in ihr zu verorten.

Wahrnehmung wird in einem neuronalen Netz modelliert durch die Aufnahme eines Vektors aus der Umgebung über die Eingabeneuronen und seine Weiterleitung an Neuronen anderer Schichten. Es findet keine Verrechnung der Eingabe statt, die die die Neuronen erregenden Reize in bedeutungsvolle Information verwandelt, sondern Bedeutung entsteht direkt durch die Interaktion der Erregungen mit den gelernten Gewichten der Synapsen. Dies kommt Gibsons Idee der direkten Wahrnehmung nahe und hilft zugleich eine Ungereimtheit in seinem Werk zu lösen, daß Wahrnehmung zwar unabhängig von Repräsentationen geschieht („die grundlegenden Angebote der Umwelt sind (...) im allgemeinen auch direkt wahrnehmbar, ohne daß ein übermäßiger Betrag an Lernen nötig wäre“, Gibson 1982, 155), aber empfänglich für Entwicklung und Lernen ist („Wahrnehmungslernen hört nie auf, solange das Leben dauert“ a. a. O., 264; vgl. auch van Wieringen 1988, 91ff).

Dabei bricht der Konnektionismus nicht mit der Idee zentraler Repräsentationen.

So we have to conclude that connectionism as a serious candidate to cognitive modelling, must be able to implement representations. A happy consequence of this conclusion is that classical descriptions become in principle available of adaptation to neural networks, and one can benefit

from the huge amount of work done in cognitive psychology, linguistics and AI [Artificial Intelligence] (Memmi 1992, 157).

Die Konstrukte generalisierter motorischer Programme und der Schemata können also, wenn auch in veränderter Form, in neuronalen Netzwerken wiedergefunden werden. Norman (1986, 536) zeichnet dabei das neue Bild eines Schemas:

Well, the schema still exists, but in somewhat different form than that which schema theorists have thought of it (...). Schemas are [now] flexible interpretive states that reflect the mixture of past experience and present circumstances. (...) Thus, the system behaves as if there were prototypical schemas, but where the prototype is constructed anew for each occasion by combining past experiences with the biases and activation levels resulting from the current experience and the context in which it occurs.

Dies ist kein großer Unterschied zu Schmidts Schemabegriff. Jedoch macht die Rechnersimulation eine Problematik in Schmidts Theorie deutlich: Die Abstraktion zu einer „Regelgeraden“ gerät bei multidimensionalen Ausgangsbedingungen und Bewegungsergebnissen zu einer kaum überwindbaren Schwierigkeit, zumal unendlich viele Parametervariationen, die auch noch nichtlinear zusammenhängen, von einer spezifischen Ausgangsbedingung zu einem spezifischen Bewegungsergebnis führen können. Vor dem Hintergrund dieser unvorstellbar großen Variabilität macht die Konstruktion von „Invarianten“ eines Programms wenig Sinn. Hier scheint doch zu sehr von phänomenal beobachtbaren Invarianten der Bewegungsklassen auf innere Vorgänge geschlossen worden zu sein. So wundert sich auch Jordan (1989, 831) über

the often fuzzy distinction made in generalized motor program theory between different generalized motor programs and different parameterizations of the same generalized motor program. The current results suggest that with respect to learning, this distinction is perhaps better treated as quantitative rather than qualitative.

In neuronalen Netzen jedenfalls lassen sich invariante „Programmelemente“ nur schwer vorstellen. Es müßte sich dabei um Neuronenkonstellationen handeln, die „fest verdrahtet“ sind und an deren Gewichten sich nichts mehr verändert. Für jede Bewegungsklasse gäbe es „feste

Verdrahtungen“, die genau bei der Bewegungsausführung feuerten. Die Variation geschähe dann durch die Aktivität weiterer Neuronen.

Eher scheint die Ansicht plausibel, daß für die Produktion von Bewegungen aus einer Klasse eine bestimmte Menge von Neuronen mit hoher Wahrscheinlichkeit erregt wird und sie ihre Erregung mit hoher Gewichtung weiterleiten, sie sind für die invarianten Bewegungsmerkmale verantwortlich. Die Variationen innerhalb der Bewegungsklasse erzeugen Neuronen, die nicht so oft und mit niedrigerer Gewichtung zur Muskelaktivierung beitragen. Dabei sind die Übergänge fließend.

### 5.3 Schwächen

Wie in dem Simulationsteil dieser Arbeit deutlich wird, gibt es auch Schwierigkeiten bei der Modellierung menschlichen Verhaltens mittels neuronaler Netze. Das dort aufgetretene Problem, daß ein Netz nach dem Erlernen einer Zielweite von beispielsweise 3 m bei einem neuen Ziel von z.B. 5 m die drei Meter reproduziert und erst nach der Fehler-rückführung die Weite verändert, ist von einiger Bedeutung. Jeder Mensch würde durch logisches Schließen darauf kommen, bei einer weiteren Zielweite kräftiger und damit weiter als zuvor zu werfen. Logisches Schließen ist aber genau die Schwäche eines neuronalen Netzes.

Dieses Problem muß nicht unlösbar sein. So ist logisches Denken keine angeborene Eigenschaft des Menschen, sondern will mühsam erlernt werden. Insofern scheint die Idee, ein künstliches Netz aus einem Dutzend Neuronen könnte mit ein paar hundert Trainingsläufen logisches Schlußfolgern lernen, anmaßend. Möglicherweise kann mit Hardware-Realisationen großer neuronaler Netze und verbesserten Lernalgorithmen über einen langen Trainingszeitraum auch logisches Schließen simuliert werden. Vielleicht stellt sich aber auch heraus, daß die Fähigkeit zum logischen Denken für die Modellierung von Bewegungsverhalten sekundär ist.

Ein diffiziles Problem stellt die Repräsentation von Zeit in einem neuro-

nalen Netz dar (vgl. Churchland & Sejnowski 1992, 379ff). Diese (auch von vielen anderen Herangehensweisen ignorierte) Schwierigkeit stellt sich bei der Simulation von Bewegungen ganz besonders, denn fast alle menschlichen Bewegungshandlungen erfahren ihre Bedeutung durch einen zeitlichen Kontext. Bullock und Grossbergs VITE codiert den zeitlichen Bewegungsverlauf, indem die Differentialgleichungen der Erregungsfunktionen der Neuronen vorgegeben werden, zeitliche Abstimmungsvorgänge mit wahrgenommenen Randbedingungen können aber nicht simuliert werden. Das dynamische Modell von Jordan & Rumelhart läßt dynamische Umgebungsaspekte außen vor, die Dynamik der Bewegung wird gelernt, indem die Parameter der vorher in das System gesteckten Bewegungsgleichung vom Netz gefunden werden. Nach Strube (1990, 137) „dürfte es unausweichlich sein, eine von der Laufzeit des Modells unabhängige Modellzeit zu definieren“. Von Simulationen in Echtzeit, wichtig für die Robotik, ist man wohl noch weit entfernt.

Eine weitere Schwäche für die Theoriebildung verdeutlicht Heuer (1992) mit folgendem Gedankenexperiment: Ein konnektionistischer Forscher modelliert in einem Netz einige simple tierische Verhaltensweisen. Dieses Modell wird Schritt für Schritt erweitert, bis das Netz exakt dem Original gleicht – und man versteht beide nicht mehr.

Daß man das Verhalten des Netzes nicht versteht, war bei dem hier konstruierten Netz aufgrund der geringen Anzahl von Neuronen und dem wohldefinierten Lernalgorithmus noch nicht der Fall, wird aber schnell zu einem Problem, wenn mehr Neuronen und kompliziertere Architekturen benutzt werden.

A pessimist might therefore not expect connectionist models to be of any use for understanding how the brain works (except for small-scale models of simple behaviors), they might rather open a new field of research: the understanding of how connectionist models work (Heuer 1992, 785).

Natürlich wäre das tiergleiche neuronale Netz ein exzellentes „Versuchskaninchen“, das ohne ethische Bedenken genauestens untersucht werden könnte, wie Heuer selbst zugibt. Er weist mit seinem Beispiel auf zwei zentrale Punkte hin:

- Im Gegensatz zu herkömmlichen Computerprogrammen kommt es dazu, daß selbst die Architekten der Netzwerke ihr Verhalten nur schwer verstehen können.
- Die Aufgabe eines Modells ist die vereinfachte Darstellung. Wenn das Modell zu kompliziert ist, verfehlt es seine eigentliche Funktion.

Trotz der Kompliziertheit der Modelle, so könnte Heuer entgegnet werden, sind aber die Prinzipien, nach denen neuronale Netzwerke funktionieren, leicht zu verstehen. Wenn durch Ausnutzung solcher Prinzipien in zugegeben komplizierten Modellen menschliches oder tierisches Verhalten simuliert werden kann, können hieraus anschauliche Gesetzmäßigkeiten formuliert werden.

In jüngster Zeit werden konnektionistische Modelle auch in der deutschen Sportwissenschaft erwähnt (Wiemeyer 1993, 1994; Hossner im Druck). Unter der Überschrift „Konnektionismus und Repräsentationalismus“ schreibt Wiemeyer (1994, 244):

Ein zentrales Defizit besteht allerdings darin, daß diese [konnektionistischen] Modelle (...) kognitive und motorische Leistungen zwar darstellen oder simulieren können, aber die dem Menschen eigene Reflexivität des Denkens (Bewußtsein) überhaupt nicht und seine Flexibilität nicht annähernd abbilden können

und stellt „als zentrales Problem“ (1993) die

Frage nach dem Originalbezug (...): Welche Merkmale des Originals „Mensch“ sollen durch konnektionistische Modelle abgebildet werden – der Aufbau menschlicher neuronaler Netzwerke oder ihre Funktion oder beides? Hier scheinen hinsichtlich beider Aspekte Verkürzungen vorgenommen zu werden (1994, 244).<sup>10</sup>

Ersteres Defizit führt schon Dörner (1991, 41) an:

Ein weiteres, bislang nur in vagen Ansätzen diskutiertes Problem einer „informationstheoretischen“ Psychologie ist das Problem der Bewußtheit und des freien Willens. (...) Hier ist allerdings theoretische Arbeit zu leisten; die oft vorgebrachte Hoffnung, daß konnektionistische Systeme gewissermaßen

<sup>10</sup>

Leider stellt Wiemeyer in seinem Vergleich zwischen „Repräsentationalismus“ und Konnektionismus weder dar, wie denn der erstere menschliches Bewußtsein abbildet, noch, wie er den verkürzten Originalbezug herstellt.

„von allein“ – wenn man sie nur komplex genug macht – so etwas wie „Bewußtsein“ produzieren würden, halte ich für ungerechtfertigt.

Zur Frage nach dem Originalbezug kann geantwortet werden, daß künstliche neuronale Netzwerke *abstrakte Modelle* des menschlichen Gehirns sind, mit denen analytische Ausschnitte aus dessen Funktion simuliert werden sollen. Verkürzungen in dieser Hinsicht sind nun einmal die Eigenschaft eines abstrakten Modells.

## 5.4 Sportpraktische Auswirkungen

Ein Ziel sportwissenschaftlicher Forschung zum motorischen Lernen und zur Bewegungskontrolle ist es, theoretisch begründete Empfehlungen für Trainings- und Übungsmethoden der Sportpraxis zu liefern. Aus den Gesetzmäßigkeiten konnektionistischer Modelle lassen sich viele Hypothesen ableiten, deren Gültigkeit im sportpraktischen Zusammenhang empirisch überprüft werden muß. Als erster Schritt auf diesem Weg wird es nötig sein, bestehende psychologische Phänomene in neuronalen Netzen abzubilden, was bisher nur ansatzweise geschehen ist. Wie modelliert der Konnektionismus Wahrnehmung, Aufmerksamkeit und Automatisierung? Wie wird eine Bewegung gesteuert, wie gelernt?

Aus dem Modell von Jordan u.a. lassen sich interessante empirische Fragestellungen ableiten. Einige Überlegungen, welche Konsequenzen zum Neulernen, zum Stabilisierungs- und Technikvariationstraining sich aus Jordans Modell ergeben, finden sich in Künzler (im Druck). Hier sei nur eine offensichtlich ins Auge springende Folgerung angeführt: Falls es nur einen quantitativen Unterschied zwischen Schmidts Programmen und Parametern gibt, müssen auch verbesserte Lernleistungen nachweisbar sein, wenn zwischen verschiedenen sportlichen Techniken variabel geübt wird. Dies muß auch positive Transferwirkungen auf andere Techniken haben.

Diese Hypothese deckt sich mit dem Alltagswissen („Alle Fußballer sind auch gute Tennisspieler“). Sie wurde in aktuellen empirischen Überprüfungen bestätigt, allerdings ohne theoretischen Bezug zu konnektionisti-



schen Modellen, sondern als Nachweis für Hossners Theorie der Motorikmodule (im Druck). Dabei konnte tendenziell festgestellt werden, daß Zielgenauigkeitsleistungen (Schlaak 1990), Einstellungen von Bewegungszeiten (Schmidt 1990; Haesner 1992) und Strategien der Geschwindigkeitsregulierung (Rollfing 1992) über verschiedene Sportarten transferierbar sind.

## 5.5 Perspektiven

Angesichts der vielen Möglichkeiten, die aus den Stärken konnektionistischer Modellbildung erwachsen, und den Herausforderungen, die seine Schwächen stellen, ist zu erwarten, daß das Modellieren mit neuronalen Netzwerken und das Erklären dieser Modelle in nächster Zukunft eine immer wichtigere Rolle in der Wissenschaft spielen wird. Ein Wechsel des erkenntnisleitenden Paradigmas (zu deutsch Modell, Beispiel: Rentsch 1989, 74) vollzieht sich nach Kuhn (1976) aber nicht allein nach logischen, sondern auch nach wissenschaftssoziologischen Gesichtspunkten. Die Tatsache, daß die Psychologie der Informationsverarbeitung „jetzt exakt so alt [ist], wie es die früheren dominierenden Richtungen gewesen waren, als es mit ihnen zu Ende ging“ (Neumann 1993, 57), also „ziemlich genau die Spanne eines Wissenschaftlerlebens“ (a.a.O., 56), veranlaßt zu Spekulationen über einen anstehenden Paradigmawechsel. In dieser Arbeit soll sich nicht weiter an ihnen beteiligt werden, zu guter letzt soll aber festgehalten bleiben, daß der Konnektionismus die beiden Merkmale zeigt, durch die Kuhn (1976, 25) ein Paradigma definiert. Der Konnektionismus ist

neuartig genug, um eine beständige Gruppe von Anhängern anzuziehen, die ihre Wissenschaft bisher auf andere Art betrieben hatten, und gleichzeitig (...) noch offen genug, um der neuen Gruppe von Fachleuten alle möglichen ungelösten Probleme zu stellen.

## Literaturverzeichnis

- Adams, J.A. (1971). A closed-loop theory of motor learning. *Journal of Motor Behavior*, 3, 111-150.
- Anderson, J.A. & Rosenfeld, E. (eds.). (1988). *Neurocomputing*. Cambridge, Mass.: The MIT Press.
- Asatryan, D.G. & Feldman, A.G. (1965). Biophysics of complex systems and mathematical models. Functional tuning of the nervous system with control of movement or maintenance of a steady posture: I. Mechanographic analysis of the work of the joint on execution of a postural task. *Biophysics*, 10, 925-935.
- Battig, W.F. (1979). The flexibility of human memory. In Cermak, L.S. & Craik, F.I.M. (eds.), *Levels of processing in human memory* (pp. 23-44). Hillsdale, NJ: Erlbaum.
- Baumann, W. (1992). Freiheitsgrade der Gelenke. In Röthig, P.H. (Hrsg.), *Sportwissenschaftliches Lexikon* (6. Aufl., S. 169). Schorndorf: Hofmann.
- Beek, P.J. & Bootsma, R.J. (1991). Physical and informational principles in modelling coordinated movements. *Human Movement Science*, 10, 81-92.
- Beek, P.J. & Meijer, O.G. (1988). On the nature of 'the' motor-action controversy. In Meijer, O.G. & Roth, K. (eds.), *Complex movement behaviour: 'The' motor-action controversy* (pp. 157-185). Amsterdam: North-Holland/Elsevier.
- Beek, P.J. & Sternad, D. (1992). *Koordination und Kontrolle von sportmotorischen Fertigkeiten aus ökospsychologischer Perspektive: Koordinative Strukturen und Information*. Papier zur 3. dvs-Sommerakademie „Ordnungs- und Organisationsleistungen menschlicher Motorik“ vom 20. bis 25. Juli 1992 in Berlin.
- Bernstein, N.A. (1967). *The coordination and regulation of movements*. New York: Pergamon Press.
- Bernstein, N.A. (1988). *Bewegungsphysiologie* (2. erw. Auflage). Leipzig: Barth.
- Borland GmbH (Hrsg.). (1990). *Turbo Pascal 6.0*. München: Borland International Inc.
- Bortz, J. (1993). *Statistik für Sozialwissenschaftler* (4. vollst. überarb. Aufl.). Berlin: Springer.
- Bullock, D. & Grossberg, S. (1988a). Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, 95, 49-90.
- Bullock, D. & Grossberg, S. (1988b). The VITE model: A neural command circuit for generating arm and articulator trajectories. In Kelso, J.A.S., Mandel, A.J. & Shlesinger, M.F. (eds.), *Dynamic patterns in complex systems* (pp. 305-326). Singapur: World Scientific.
- Bullock, D. & Grossberg, S. (1988c). Self-organizing neural architectures for eye movements, arm movements, and eye-arm coordination. In Haken, H. (ed.), *Neural and synergetic computers* (pp. 197-228). Berlin: Springer.
- Bullock, D. & Grossberg, S. (1989). VITE and FLETE: Neural modules for trajectory formation and postural control. In Hershberger (ed.), *Volitional action* (pp. 253-297). Amsterdam: North-Holland/Elsevier.
- Bullock, D. & Grossberg, S. (1990). Spinal network computations enable independent control of muscle length and joint compliance. In Eckmiller, R. (ed.), *Advanced neural computers* (pp. 349-356). Amsterdam: North-Holland/Elsevier.
- Bullock, D. & Grossberg, S. (1991a). Adaptive neural networks for control of movement trajectories invariant under speed and force rescaling. *Human Movement Science*, 10, 3-54.
- Bullock, D. & Grossberg, S. (1991b). Reply to comments on 'Adaptive neural networks for control of movement trajectories invariant under speed and force rescaling'. *Human Movement Science*, 10, 133-158.

- Churchland, P.S. & Sejnowski, T.J. (1992). *The computational brain*. Cambridge, Mass.: The MIT Press.
- Clark, A. & Lutz, R. (eds.). (1992). *Connectionism in context*. London: Springer.
- Creutzfeld, O. (1991). „Biologisches Verständnis der Hirnfunktion“ Teil (B): Hirnorganisation. In Der Bundesminister für Forschung und Technologie (Hrsg.), *Neurobiologie/Hirnforschung - Neuroinformatik, Künstliche Intelligenz* (S. 45-50). Bonn: BMFT.
- Daish, C.B. (1972). *The physics of ball games*. London: Engl. Universities Press.
- Deno, D.C. (1991). Review of 'Adaptive neural networks for control of movement trajectories invariant under speed and force rescaling'. *Human Movement Science*, 10, 73-80.
- Dörner, D. (1991). Psychologie, „Künstliche Intelligenz“ und Konnektionismus. In Der Bundesminister für Forschung und Technologie (Hrsg.), *Neurobiologie/Hirnforschung - Neuroinformatik, Künstliche Intelligenz* (S. 38-43). Bonn: BMFT.
- Feldman, A.G. (1966a). Functional tuning of the nervous system with control of movement or maintenance of a steady posture: II. Controllable parameters of the muscles. *Biophysics*, 11, 565-578.
- Feldman, A.G. (1966b). Functional tuning of the nervous system with control of movement or maintenance of a steady posture: III. Mechanographic analysis of the execution by man of the simplest motor tasks. *Biophysics*, 11, 667-675.
- Feldman, A.G. (1986). Once more on the equilibrium point hypothesis (model) for motor control. *Journal of Motor Behavior*, 18, 17-54.
- Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-391.
- Flash, T. & Hogan, N. (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 5, 1688-1703.
- Fodor, J.A. (1975). *The language of thought*. Cambridge, MA.: Harvard University Press.
- Fodor, J.A. & Pylyshyn, Z.W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28, 3-71.
- Gaudiano, P. & Grossberg, S. (1992). Adaptive vector integration to endpoint: Self-organizing neural circuits for control of planned movement trajectories. *Human Movement Science*, 11, 141-155.
- Georgopoulos, A.P., Kalaska, J.F. & Massey, J.T. (1981). Spatial trajectories and reaction times of aimed movements: Effects of practice, uncertainty, and change in target location. *Journal of Neurophysiology*, 46, 725-743.
- Gibson, J.J. (1982). *Wahrnehmung und Umwelt*. München: Urban & Schwarzenberg.
- Grossberg, S. (1982). *Studies of mind and brain: Neural principles of learning, perception, development, cognition and motor control*. Boston: Reidel Press.
- Grossberg, S. & Kuperstein, M. (1986). *Neural dynamics of adaptive sensory-motor control: Ballistic eye movements*. Amsterdam: North-Holland/Elsevier.
- Haas, R., Fuchs, A., Haken, H., Horvath, E., Pandya, A. & Kelso, J.A.S. (1995). Recognition of dynamic patterns by a synergetic computer. In Kruse, P. & Stadler, M. (eds.), *Ambiguity in mind and nature* (pp. 139-155). Berlin: Springer.
- Haesner, M. (1992). *Zur Optimierung der Bewegungssteuerung im leichtathletischen Sprint*. Unveröffentlichte Examensarbeit, Freie Universität, Berlin.
- Haken, H. (1991). Synergetics of movement coordination. *Human Movement Science*, 10, 113-116.
- Haken, H. (1992). Synergetik: Von der Musterbildung zur Mustererkennung. *Nova acta Leopoldina NF 67, Nr 281*, 53-71.

- Haken, H., Kelso, J.A.S. & Bunz, H. (1985). A theoretical model of phase transitions in human hand movements. *Biological Cybernetics*, 51, 347-356.
- Hanson, S.J. & Olson, C.R. (1990). Introduction: Connectionism and neuroscience. In Hanson, S.J. & Olson, C.R. (eds.), *Connectionist modeling and brain function: The developing interface* (pp. 1-4). Cambridge, Mass.: The MIT Press.
- Hebb, D.O. (1949). *The organisation of behavior*. New York: Wiley.
- Helm, G. (1991). *Symbolische und konnektionistische Modelle der menschlichen Informationsverarbeitung*. Berlin: Springer.
- Heuer, H. (1983). *Bewegungslernen*. Stuttgart: Kohlhammer.
- Heuer, H. (1985). Wie wirkt mentale Übung? *Psychologische Rundschau*, 36, 191-200.
- Heuer, H. (1991). Some characteristics of VITE. *Human Movement Science*, 10, 55-64.
- Heuer, H. (1992). Computations, neural networks and the limits of human understanding. *Behavioral and Brain Sciences*, 15, 758.
- Hinton, G.E. (1984). Parallel computations for controlling an arm. *Journal of Motor Behavior*, 16, 171-194.
- Hirayama, M., Kawato, M. & Jordan, M.I. (1993). The cascade neural network model and a speed-accuracy trade-off of arm movements. *Journal of Motor Behavior*, 25, 162-174.
- Holzkamp, K. (1993). *Lernen. Subjektwissenschaftliche Grundlegung*. Frankfurt/Main: Campus.
- Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79, 2554-2558.
- Horak, M. (1992). The utility of connectionism for motor learning: A reinterpretation of contextual interference in movement schemas. *Journal of Motor Behavior*, 24, 58-66.
- Hossner, E.-J. (1991). „Kennst Du die Grenzen, kennst Du den Weg!“ *sportpsychologie*, 5(1), 11-16.
- Hossner, E.-J. (im Druck). *Module der Motorik – Bausteine des Bewegens*. Schorndorf: Hofmann.
- Hossner, E.-J., Munzert, J. & Olivier, N. (1995). Bewegungslernen als theoretisches und sportpraktisches Problem. In Rode, J. & Phillip, H. (Hrsg.), *Sport in Schule, Verein und Betrieb*. (Bericht über den 11. Sportwissenschaftlichen Hochschultag der dvs vom 22.-24. 9. 1993 in Potsdam, S. 221-226). Sankt Augustin: Academia.
- Hucho, F. (1991). Biologisches Verständnis der Hirnfunktion Teil (A): Das Neuron. In Der Bundesminister für Forschung und Technologie (Hrsg.), *Neurobiologie/Hirnforschung - Neuroinformatik, Künstliche Intelligenz* (S. 43-45). Bonn: BMFT.
- James, W. (1990). *Psychology (Briefer Course)*. Auszugsweise neu abgedruckt in Anderson, J.A. & Rosenfeld, E. (eds.). (1988), *Neurocomputing* (pp. 4-15). Cambridge, Mass.: The MIT Press.
- Jordan, M.I. (1989). Motor learning and the degrees of freedom problem. In Jeannerod, M. (ed.), *Attention and performance XIII* (pp. 796-836). Hillsdale, NJ.: Erlbaum.
- Jordan, M.I. (1992). Constrained supervised learning. *Journal of Mathematical Psychology*, 36, 396-425.
- Jordan, M.I. & Rosenbaum, D.A. (1989). Action. In Posner, M.I. (ed.), *Foundations of cognitive science* (pp. 727-767). Cambridge, Mass.: The MIT Press.
- Jordan, M.I. & Rumelhart, D.E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16, 307-354.
- Kawato, M. (1990a). Computational schemes and neural network models for formation and control of multijoint arm trajectory. In Miller W.T., Sutton, R.S. & Werbos, P.J. (eds.), *Neural networks for control* (pp. 197-228). Cambridge, Mass.: The MIT Press.

- Kawato, M. (1990b). Feedback-error-learning neural network for supervised motor learning. In Eckmiller, R. (ed.), *Advanced neural computers* (pp. 365-372). Amsterdam: North-Holland/Elsevier.
- Kawato, M., Furukawa, K. & Suzuki, R. (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57, 169-185.
- Kawato, M., Maeda, Y., Uno, Y. & Suzuki, R. (1990). Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion. *Biological Cybernetics*, 62, 257-288.
- Keele, S.W. (1968). Movement control in skilled motor performance. *Psychological Bulletin*, 70, 387-403.
- Keele, S.W. & Jennings, P.J. (1992). Attention in the representation of sequence. *Human Movement Science*, 11, 125-138.
- Keller, J. (1990). Konnektionismus - ein neues Paradigma zur Wissensrepräsentation? *Linguistische Berichte*, 128, 298-332.
- Kelso, J.A.S. (1977). Motor control mechanisms underlying human movement production. *Journal of Experimental Psychology: Human Perception and Performance*, 3, 529-543.
- Kelso, J.A.S. (1991). Anticipatory dynamical systems, intrinsic pattern dynamics and skill learning. *Human Movement Science*, 10, 93-112.
- Kohonen, T. (1984). *Self-organization and associative memory*. Berlin: Springer.
- Kugler, P.N., Kelso, J.A.S. & Turvey, M.T. (1980). On the concept of coordinative structures as dissipative structures: I. Theoretical lines of convergence. In Stelmach, G.E. & Requin, J. (eds.), *Tutorials in motor behavior* (pp. 3-47). Amsterdam: North-Holland/Elsevier.
- Kuhn, T.S. (1976). *Die Struktur wissenschaftlicher Revolutionen* (2. überarb. Aufl.). Frankfurt/Main: Suhrkamp.
- Künzell, S. (im Druck). Neuronale Netzwerke: Modelle für kognitive Prozesse der Bewegungsproduktion? In Bundesinstitut für Sportwissenschaft (Hrsg.), *Neue Forschungs- und Auswertetechnologien im Sport*. Schorndorf: Hofmann.
- Lames, M. (1991). *Die komplexe sportliche Leistung - ein nichtlineares dynamisches System?* Papier zum Symposium „Synergetik und Systeme in Training und Wettkampf“ vom 4.-6. 12. 1991 in Kiel.
- Lashley, K.S. (1917). The accuracy of movement in the absence of excitation from the moving organ. *The American Journal of Physiology*, 43, 169-194.
- Lashley, K.S. (1950). In search of the engram. In *Society of Experimental Biology Symposium No. 4: Psychological mechanisms in animal behavior* (pp. 478-505). London: Cambridge University Press.
- Le Cun, Y. (1985). Une procédure d'apprentissage pour réseau à seuil asymétrique. *Cognitiva 85: A la frontière de l'intelligence artificielle des sciences de la connaissance des neurosciences*, 599-604.
- Lee, D.N. (1976). A theory of visual control of braking based on information about time to collision. *Perception*, 5, 437-459.
- Magill, R.A. & Hall, K.G. (1990). A review of the contextual interference effect in motor skill acquisition. *Human Movement Science*, 9, 241-289.
- Marr, D. (1982). *Vision*. San Francisco: Freeman.
- Marteniuk, R.G. (1992). Issues in goal directed motor learning: Feedforward control, motor equivalence, specificity, and artificial neural networks. In Stelmach, G.E. & Requin, J. (eds.), *Tutorials of motor behavior II* (pp. 101-124). Amsterdam: North-Holland/Elsevier.
- Masson, M.E.J. (1990). Cognitive theories of skill acquisition. *Human Movement Science*, 9, 221-239.

- McClelland, J.L., Rumelhart, D.E. & Hinton, G.E. (1986). The appeal of parallel distributed processing. In Rumelhart, D.E., McClelland, J.L. & the PDP research group (eds.), *Parallel distributed processing. Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 3-44). Cambridge, Mass.: The MIT Press.
- McClelland, J.L., Rumelhart, D.E. & the PDP Research Group (eds.). (1986). *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 2: Psychological and biological models*. Cambridge, Mass.: The MIT Press.
- McCulloch, W.S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.
- Meijer, O.G. & Roth, K. (eds.). (1988). *Complex movement behaviour: 'The' motor-action controversy*. Amsterdam: North-Holland/Elsevier.
- Memmi, D. (1992). Connectionism and artificial intelligence as cognitive models. In Clark, A. & Lutz, R. (eds.), *Connectionism in context* (pp. 145-166). London: Springer.
- Minsky, M.L. & Papert, S.A. (1969). *Perceptrons: An introduction to computational geometry*. Cambridge, Mass.: The MIT Press.
- Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research*, 42, 223-227.
- Morasso, P. (1990). Neural representation of motor synergies. In Eckmiller, R. (ed.), *Advanced neural computers* (pp. 51-59). Amsterdam: North-Holland/Elsevier.
- Morasso, P. (1992). Neural mechanisms of synergy formation. *Human Movement Science*, 11, 169-180.
- Morasso, P. & Sanguineti, V. (1992). Neural models of distributed motor control. In Stelmach, G.E. & Requin, J. (eds.), *Tutorials in motor behavior II* (pp. 3-30). Amsterdam: North Holland/Elsevier.
- Moxley, S.E. (1979). Schema: The variability of practice hypothesis. *Journal of Motor Behavior*, 11, 65-70.
- Munzert, J. (1992). Motorik-Repräsentation, Bewegungswissen und Bewegungshandeln. *Sportwissenschaft*, 22, 344-356.
- Neumann, O. (1985). Informationsverarbeitung, Künstliche Intelligenz und die Perspektiven der Kognitionspsychologie. In Neumann, O. (Hrsg.), *Perspektiven der Kognitionspsychologie* (S. 3-37). Berlin: Springer.
- Neumann, O. (1993). Psychologie der Informationsverarbeitung: Aktuelle Tendenzen und einige Konsequenzen für die Aufmerksamkeitsforschung. In Daus, R. & Blischke, K. (Hrsg.), *Aufmerksamkeit und Automatisierung in der Bewegungsforschung* (S. 56-78). St. Augustin: Academia.
- Neumann, O. & Prinz, W. (1990). Prologue: Historical approaches to perception and action. In Neumann, O. & Prinz, W. (eds.), *Relationship between perception and action* (pp. 5-19). Berlin: Springer.
- Norman, D.A. (1986). Reflections on cognition and parallel distributed processing. In McClelland, J.L., Rumelhart, D.E. & the PDP research group (eds.), *Parallel distributed processing. Explorations in the microstructure of cognition. Vol. 2: Psychological and biological models* (pp. 531-546). Cambridge, Mass.: The MIT Press.
- Parker, D.B. (1985). *Learning-logic*. Cambridge, Mass.: The MIT Press.
- Parks, R.W., Long, D.L., Levine, D.S., Crockett, D.J., McGeer, E.G., McGeer, P.L., Dalton, I.E., Zec, R.F., Becker, R.E., Coburn, K.L., Siler, G., Nelson, M.E. & Bower, J.M. (1991). Parallel distributed processing and neural networks: Origins, methodology and cognitive functions. *International Journal of Neuroscience*, 60, 195-214.
- Pitts, W. & McCulloch, W.S. (1947). How we know the universals: The perception of auditory and visual forms. *Bulletin of Mathematical Biophysics*, 9, 127-147.

- Rentsch, T. (1989). Paradigma. In Ritter, J. & Grunder, M. (Hrsg.), *Historisches Wörterbuch der Philosophie, Band 7*. Basel: Schwabe.
- Rojas, R. (1993). *Theorie der neuronalen Netze. Eine systematische Einführung*. Berlin: Springer.
- Rollfing, M. (1992). *Zur zeitlichen Strukturierung sportbezogener Bewegungstechniken*. Unveröffentlichte Examensarbeit, Freie Universität, Berlin.
- Rosenbaum, D.A. (1983). The movement precuing technique: Assumptions, applications and extensions. In Magill, R.A. (ed.), *Memory and control of action* (pp. 231-274). Amsterdam: North-Holland/Elsevier.
- Rosenbaum, D.A. (1991). *Human motor control*. San Diego: Academic Press.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organisation in the brain. Neugedruckt in Anderson, J.A. & Rosenfeld, E. (eds.). (1988). *Neurocomputing* (pp. 92-114). Cambridge, Mass.: The MIT Press.
- Roth, K. (1989). *Taktik im Sportspiel*. Schorndorf: Hofmann.
- Roth, K. (1991). „Erst das Leichte, dann das Schwere - stufenweise richtig lehre!“ *sportpsychologie*, 5(1), 5-10.
- Roth, K. & Raab, M. (1994). *Taktische Regelbildung: „Mühsam, konzentriert, intentional oder mühe-los, nebensächlich, inzidentell?“* Vortrag gehalten auf dem Symposium „Empirische Forschung im Sportspiel: Methodologie, Fakten und Reflexionen“ vom 15.-17. Juni 1994 in Kiel.
- Rumelhart, D.E., Hinton, G.E. & Williams, D.E. (1986). Learning internal representations by error propagation. In Rumelhart, D.E., McClelland, J.L. & the PDP research group (eds.), *Parallel distributed processing. Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 318-362). Cambridge, Mass.: The MIT Press.
- Rumelhart, D.E. & McClelland, J.L. (1986). PDP models and general issues in cognitive science. In Rumelhart, D.E., McClelland, J.L. & the PDP research group (eds.), *Parallel distributed processing. Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 110-146). Cambridge, Mass.: The MIT Press.
- Rumelhart, D.E., McClelland, J.L. & the PDP Research Group (eds.). (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations*. Cambridge, Mass.: The MIT Press.
- Rumelhart, D.E. & Norman, D.A. (1982). Simulating a skilled typist: A study of skilled cognitive-motor performance. *Cognitive Science*, 6, 1-36.
- Rutkowska, J.C. (1992). Action, connectionism and enaction: A developmental perspective. In Clark, A. & Lutz, R. (eds.), *Connectionism in context* (pp. 17-36). London: Springer.
- Scheerer, E. (1993). Mentale Repräsentation in interdisziplinärer Perspektive. *Zeitschrift für Psychologie*, 201, 136-166.
- Schlaak, C. (1990). *Zum Parametertransfer im Sportspiel*. Unveröffentlichte Examensarbeit, Freie Universität, Berlin.
- Schmidt, G. (1990). *Zum Transfer von Bewegungsparametern bei zyklischen Bewegungen im Sport*. Unveröffentlichte Examensarbeit, Freie Universität, Berlin.
- Schmidt, R.A. (1975/1990). A schema theory of discrete motor skill learning. *Psychological Review* 82 (1975), 225-260. Deutsche Übersetzung in Kaul, P. & Zimmermann, K.W. (Hrsg. und Übers.), *Psychomotorik in Forschung und Praxis Bd. 2*, Kassel (1990): Gesamt-Hochschulbibliothek Kassel.
- Schmidt, R.A. (1988a). Motor and action perspectives on motor behaviour. In Meijer, O.G. & Roth, K. (eds.), *Complex movement behaviour: 'The' motor-action controversy* (pp. 3-44). Amsterdam: North Holland/Elsevier.
- Schmidt, R.A. (1988b). *Motor control and learning* (2<sup>nd</sup> rev. ed.). Champaign, Ill.: Human Kinetics Publishers.

- Schmidt, R.C., Treffner, P.J. & Turvey, M.T. (1991). Neural networks and the first and second rounds of theorizing on Bernstein's problem. *Human Movement Science*, 10, 117-132.
- Schneider, K. (1992). *Ordnungs- und Organisationsleistungen menschlicher Motorik*. Papier zur 3. dvs-Sommerakademie „Ordnungs- und Organisationsleistungen menschlicher Motorik“ vom 20. bis 25. Juli 1992 in Berlin.
- Schneider, K., Zernicke, R.F., Schmidt, R.A. & Hart, T.J. (1989). Changes in limb dynamics during the practice of rapid arm movements. *Journal of Biomechanics*, 22, 805-817.
- Schneider, K., Zernicke, R.F., Ulrich, B.D., Jensen, J.L. & Thelen, E. (1990). Understanding movement control in infants through the analysis of limb intersegmental dynamics. *Journal of Motor Behavior*, 22, 493-520.
- Schneider, W. (1987). Connectionism: Is it a paradigm shift for psychology? *Behavior Research Methods, Instruments, & Computers*, 19, 73-83.
- Schöner, G., Haken, H. & Kelso, J.A.S. (1986). A stochastic theory of phase transitions in human hand movement. *Biological Cybernetics*, 53, 247-257.
- Schöner, G., Zanone, P.G. & Kelso, J.A.S. (1992). Learning as change of coordination dynamics: Theory and experiment. *Journal of Motor Behavior*, 24, 29-48.
- Shea, J.B. & Morgan, R.L. (1979). Contextual interference effects on the acquisition, retention, and transfer of a motor skill. *Journal of Experimental Psychology: Human Learning and Memory*, 5, 179-187.
- Smolensky, P. (1988). On the proper treatment of connectionism. *Behavior and Brain Science*, 11, 1-74.
- Spencer, H. (1903, Original 1. Aufl. 1855). *Die Principien der Psychologie*. Stuttgart: Schweizerbart.
- Stelmach, G.E. & Requin, J.H. (1992). *Tutorials in motor behavior II*. Amsterdam: North-Holland/Elsevier.
- Strube, G. (1990). Neokonnektionismus: Eine neue Basis für die Theorie und Modellierung menschlicher Kognition? *Psychologische Rundschau*, 41, 129-143.
- Sutherland, S. (1989). *MacMillan Dictionary of Psychology*. London/Basingstoke: MacMillan.
- Turvey, M.T. (1977). Preliminaries to a theory of action with reference to vision. In Shaw, R. & Bransford, J. (eds.), *Perceiving, acting, and knowing: Toward an ecological psychology* (pp. 211-265). Hillsdale, NJ: Erlbaum.
- Uno, Y., Kawato, M. & Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement. Minimum torque-change model. *Biological Cybernetics*, 61, 89-101.
- Van Rossum, J.H.A. (1987). *Motor development and practice: The variability of practice hypothesis in perspective*. Amsterdam: Free University Press.
- Van Rossum, J.H.A. (1990). Schmidt's schema theory: The empirical base of the variability of practice hypothesis. *Human Movement Science*, 9, 387-435.
- Van Wieringen, P.C.W. (1988). Kinds and levels of explanation: Implications for the motor systems versus action systems controversy. In Meijer, O.G. & Roth, K. (eds.), *Complex movement behaviour: The motor-action controversy* (pp. 87-120). Amsterdam: North-Holland/Elsevier.
- Vereijken, B., van Emmerik, R.E.A., Whiting, H.T.A. & Newell, K.M. (1992). Free(z)ing degrees of freedom in skill acquisition. *Journal of Motor Behavior*, 24, 133-142.
- Walker, S.F. (1992). A brief history of connectionism and its psychological implications. In Clark, A. & Lutz, R. (eds.), *Connectionism in context* (pp. 123-144). London: Springer.
- Wiemeyer, J. (1992). Motorische Kontrolle und motorisches Lernen im Sport. *sportpsychologie*, 6(1), 5-11; (2), 5-12.



- Wiemeyer, J. (1993). *Auf der Suche nach einem adäquaten Verständnis der Bewegung des Menschen*. Poster zum 11. Sportwissenschaftlichen Hochschultag der dvs „Sport in Schule, Verein und Betrieb“ vom 22. - 24. 9. 1993 in Potsdam.
- Wiemeyer, J. (1994). Interne Bewegungsrepräsentationen. *Sportwissenschaft*, 24, 233-235.
- Wilberg, R.B. (1991). Adaptive neural networks in movement control: Some persistent concerns. *Human Movement Science*, 10, 65-72.
- Willimczik, K. & Roth, K. (1983). *Bewegungslehre*. Reinbek: Rohwolt.
- Zelaznik, H.N., Schmidt, R.A. & Gielen, S.C.A.M. (1986). Kinematic properties of rapid aimed hand movements. *Journal of Motor Behavior*, 18, 353-372.

## Anhang: Quellcode des erstellten Programms

Das Programm wurde in Turbo-Pascal Version 6.0 der Firma Borland (1990) geschrieben. Nähere Erläuterungen finden sich in Kapitel 4.3.

UNIT datastru;

INTERFACE

```
CONST maxdim = 6;
      num_forward_vectors = 500;
      gamma = 1;
      num_uebungen = 4;
      g = 9.81;
      delta_h = 1;
      width: ARRAY[1..7] OF REAL = (3, 5, 7, 9, 4, 8, 10);

TYPE neuronT = ARRAY [1..maxdim] OF REAL;
   layerT = ARRAY [1..maxdim] OF neuronT;
   layercatT = (input, hidden, output, error);
   dimT = ARRAY [input..output] OF INTEGER;
   netT = RECORD
       dim: dimT;
       layer: ARRAY [hidden..output] OF layerT;
   END;
   parameterT = (vnull, alfa, sxmat, sxnet);
   environ_vectorT = ARRAY[vnull..sxnet] OF REAL;
   forward_model_dataT = ARRAY [1..num_forward_vectors] OF environ_vectorT;
   vectorT = ARRAY [1..maxdim] OF REAL;
```

IMPLEMENTATION

END

---

UNIT data;

INTERFACE

```
USES datastru;
CONST delta_h = 1;
PROCEDURE init_forward_model_data;
PROCEDURE norm_vector (VAR vec: environ_vectorT);
PROCEDURE unnorm_vector (VAR vec: environ_vectorT);
PROCEDURE write_data (filename: STRING; limit: INTEGER);
PROCEDURE fetch_vector (i: INTEGER; VAR vec: environ_vectorT);
PROCEDURE put_vector (i: INTEGER; VAR vec: environ_vectorT);
FUNCTION compute_distance (v, a: REAL):REAL;
```

IMPLEMENTATION

VAR fdata: forward\_model\_dataT;

FUNCTION compute\_distance (v, a: REAL):REAL;

VAR rad: REAL;

{berechnet aus der Geschwindigkeit v und dem Winkel a die Weite zum Korb als Funktionsergebnis. Falls der Korb nicht erreicht werden kann, wird 0 als Funktionsergebnis zurueckgeliefert }

BEGIN

rad := a \* pi / 180;

IF sqr (sin(rad) \* v) < 2 \* delta\_h \* g THEN compute\_distance := 0

ELSE compute\_distance := (cos(rad) \* v/g) \*(sqrt (sqr (sin (rad) \* v) - 2 \* delta\_h \* g) + sin(rad) \* v);

END;

```

PROCEDURE init_forward_model_data;
VAR i: INTEGER;
    test: REAL;
BEGIN
    FOR i := 1 TO num_forward_vectors DO BEGIN
        fddata[i,vnull] := (random * 4 + 7);
        {geschwindigkeiten zwischen 7 und 11 m/s}
        fddata[i,alfa] := (random * 20 + 40);
        {Winkel zwischen 40 und 60 Grad}
        fddata[i,sxmat] := compute_distance (fddata[i,vnull], fddata[i,alfa]);
        IF fddata[i,sxmat] = 0 {Korb unerreichbar} THEN dec (i);
    END;
END;

```

```

PROCEDURE norm_vector (VAR vec: environ_vectorT);
BEGIN
    vec[vnull] := (vec[vnull] - 7) / (4 * 1.25) + 0.1;
    vec[alfa] := (vec[alfa] - 40) / (20 * 1.25) + 0.1;
    vec[sxmat] := (vec[sxmat] - 2.5) / 9;
    vec[sxnet] := (vec[sxnet] - 2.5) / 9;
    IF vec[sxnet] < 0 {fuer Eingaben, die den Korb nicht erreichen}
        THEN vec[sxnet] := 0.00001;
    IF vec[sxmat] < 0 {fuer Eingaben, die den Korb nicht erreichen}
        THEN vec[sxmat] := 0.00001;
END;

```

```

PROCEDURE unnorm_vector (VAR vec: environ_vectorT);
BEGIN
    vec[vnull] := (vec[vnull] - 0.1) * 4 * 1.25 + 7;
    vec[alfa] := (vec[alfa] - 0.1) * 20 * 1.25 + 40;
    vec[sxmat] := vec[sxmat] * 9 + 2.5;
    vec[sxnet] := vec[sxnet] * 9 + 2.5;
END;

```

```

PROCEDURE write_data (filename: STRING; limit: INTEGER);
VAR f: TEXT;
    i: INTEGER;
    j: parameterT;
BEGIN
    assign (f, filename);
    rewrite (f);
    FOR i := 1 TO limit DO BEGIN
        FOR j := vnull TO sxnet DO
            write (f, fddata[i,j]:6:4, ' ');
        writeln (f);
    END;
    close (f);
END;

```

```

PROCEDURE fetch_vector (i: INTEGER; VAR vec: environ_vectorT);
BEGIN
    vec := fddata[i];
END;

```

```

PROCEDURE put_vector (i: INTEGER; VAR vec: environ_vectorT);
BEGIN
    fdata[i] := vec;
END;

BEGIN
    randomize;
END.

```

---

```

UNIT nets;

INTERFACE
    USES data, datastru;

    PROCEDURE transform_vector (v1: environ_vectorT; VAR fl, di: vectorT);
    PROCEDURE feedforward (net: netT; invector: vectorT; VAR outvector: vectorT);
    PROCEDURE learn1 (invector: vectorT; target: vectorT; VAR net: netT;
        VAR outvector: vectorT; gamma: REAL);
    PROCEDURE learn2 (VAR profile: TEXT; data: environ_vectorT; forward_net: netT;
        VAR inverse_net: netT; gamma: REAL);
    PROCEDURE test_inverse_net (invector: environ_vectorT; pnet, fnet: netT;
        VAR outvector: environ_vectorT);
    PROCEDURE init_net (VAR net: netT; dim: dimT);
    PROCEDURE read_net (VAR net: netT; filename: STRING);
    PROCEDURE write_net (net: netT; filename: STRING);

IMPLEMENTATION

CONST c = 1;

VAR out1: vectorT; {wird von feedforward per Seiteneffekt ein Wert
                    zugewiesen, der auch von learn benutzt werden muss}

PROCEDURE init_net;
VAR i, j: INTEGER;
    l: layercatT;
BEGIN
    FOR l := hidden TO output DO BEGIN
        FOR i := 1 TO dim[l] DO BEGIN
            FOR j := 1 TO dim[pred(l)]+1 DO
                net.layer[l,i,j] := random - 0.5;
            {initialisierung mit werten zwischen -0,5 und 0,5}
        END;
    END;
    net.dim := dim;
END;

PROCEDURE read_net;
VAR f: file OF netT;
BEGIN
    assign (f,filename);
    reset (f);
    read (f, net);
    close (f);
END;

```

```

PROCEDURE write_net;
VAR f: file OF netT;
BEGIN
    assign (f,filename);
    rewrite (f);
    write (f, net);
    close (f);
END;

FUNCTION fire (neuron: neuronT; erregung: vectorT; dim: INTEGER): REAL;
VAR sum: REAL;
    i: INTEGER;
BEGIN
    sum := 0;
    FOR i := 1 TO dim DO
        sum := sum + erregung[i] * neuron[i];
    fire := 1/(1+exp((0-sum)*c));
END;

PROCEDURE pass (layer: layerT; invector: vectorT; dimin, dimout: INTEGER;
    VAR result: vectorT);
VAR i: INTEGER;
BEGIN
    invector[dimin+1] := 1; {schwollenwert}
    FOR i := 1 TO dimout DO result[i] := fire (layer[i], invector, dimin+1);
END;

PROCEDURE feedforward; {seiteneffekt auf das globale out1}
VAR layer: layerT;
BEGIN
    pass (net.layer[hidden], invector, net.dim[input], net.dim[hidden], out1);
    pass (net.layer[output], out1, net.dim[hidden], net.dim[output], outvector);
END;

PROCEDURE correction (delta, invector: vectorT; dim1, dim2: INTEGER;
    gamma: REAL; VAR layer: layerT);
VAR neuron1, weight1: INTEGER;
BEGIN
    invector [dim1+1] := 1; {schwollenwert}
    FOR neuron1 := 1 TO dim2 DO BEGIN
        FOR weight1 := 1 TO dim1+1 DO
            layer[neuron1,weight1] := layer[neuron1,weight1]+
                invector[weight1]*delta[neuron1]*gamma;
        END;
    END;
END;

PROCEDURE compute_delta1 (out1, delta2: vectorT; dim1, dim2: INTEGER;
    layer2: layerT; VAR result: vectorT);
VAR neuron1i, neuron2i: INTEGER;
    sum: REAL;
BEGIN
    FOR neuron1i := 1 TO dim1 DO BEGIN
        sum := 0;
        FOR neuron2i := 1 TO dim2 DO
            sum := sum + layer2[neuron2i,neuron1i]*delta2[neuron2i];
        result[neuron1i] := out1[neuron1i]*(1.0-out1[neuron1i])*sum;
    END;
END;

```

```

PROCEDURE compute_delta2
    (out2, target: vectorT; dim: INTEGER; VAR result: vectorT);
VAR i: INTEGER;
BEGIN
    FOR i := 1 TO dim DO
        result[i] := (target[i] - out2[i])*(1 - out2[i])*out2[i];
    END;
END;

PROCEDURE learn1 (invector, target: vectorT; VAR net: netT;
    VAR outvector: vectorT; gamma: REAL);
VAR delta1, delta2: vectorT;
BEGIN
    feedforward (net, invector, outvector);
    compute_delta2 (outvector, target, net.dim[output], delta2);
    compute_delta1 (out1, delta2, net.dim[hidden], net.dim[output], net.layer[output],
        delta1);
    correction (delta2, out1, net.dim[hidden], net.dim[output],
        gamma, net.layer[output]);
    correction (delta1, invector, net.dim[input], net.dim[hidden],
        gamma, net.layer[hidden]);
END;

PROCEDURE learn2 (VAR profile: TEXT; data: environ_vectorT; forward_net: netT;
    VAR inverse_net: netT; gamma: REAL);
VAR out2, out3, outh: vectorT;
    outvector, flight, distance, delta1, delta2, delta3, delta4: vectorT;
    protvec: environ_vectorT;
BEGIN
    norm_vector (data);
    transform_vector (data, flight, distance);
    feedforward (inverse_net, distance, out2);
    outh := out1; {wg. Seiteneffekt!}
    feedforward (forward_net, out2, outvector);
    out3 := out1;
    out1 := outh;
    protvec[vnull] := out2[1];
    protvec[alfa] := out2[2];
    protvec[sxmat] := distance[1];
    protvec[sxnet] := 0;
    unnorm_vector (protvec);
    protvec[sxnet] := compute_distance (protvec[vnull], protvec[alfa]);
    writeln (profile, 'vnull: ', protvec[vnull]:6:4, ' alfa: ', protvec[alfa]:6:4,
        ' sx*: ', protvec[sxmat]:6:4, ' sx: ', protvec[sxnet]:6:4);
    norm_vector (protvec);
    outvector[1] := protvec[sxnet];
    compute_delta2 (outvector, distance, forward_net.dim[output], delta4);
    compute_delta1 (out3, delta4, forward_net.dim[hidden], forward_net.dim[output],
        forward_net.layer[output], delta3);
    compute_delta1 (out2, delta3, forward_net.dim[input], forward_net.dim[hidden],
        forward_net.layer[hidden], delta2);
    compute_delta1 (out1, delta2, inverse_net.dim[hidden], inverse_net.dim[output],
        inverse_net.layer[output], delta1);
    correction (delta2, out1, inverse_net.dim[hidden], inverse_net.dim[output],
        gamma, inverse_net.layer[output]);
    correction (delta1, distance, inverse_net.dim[input], inverse_net.dim[hidden],
        gamma, inverse_net.layer[hidden]);
END;

```

```
PROCEDURE transform_vector (v1: environ_vectorT; VAR f1, d1: vectorT);
BEGIN
```

```
    f1[1] := v1[vnull];
    f1[2] := v1[alfa];
    d1[1] := v1[sxmat];
```

```
END;
```

```
PROCEDURE test_inverse_net (invector: environ_vectorT; pnet, fnet: netT;
    VAR outvector: environ_vectorT);
```

```
VAR flight, distance, out2, out4: vectorT;
```

```
BEGIN
```

```
    outvector[sxmat] := invector[sxmat];
    transform_vector (invector, flight, distance);
    feedforward (pnet, distance, out2);
    outvector[vnull] := out2[1];
    outvector[alfa] := out2[2];
    unnorm_vector (outvector);
    outvector[sxnet] := compute_distance (outvector[vnull], outvector[alfa]);
```

```
END;
```

```
{main}
```

```
END;
```

---

```
program connectionist_model;
```

```
USES nets, data, datastru;
```

```
CONST
```

```
    number_of_inverse_iterations = 800;
    number_of_forward_iterations = 2000;
    number_of_tasks = 4; {kleiner als num_iter!}
    num_versuch = '1';
```

```
VAR forward_net, inverse_net: netT;
```

```
    s1, s2, s3, vnet: STRING[3];
```

```
    gruppe, i, neti: INTEGER;
```

```
    datavector, test_vector: environ_vectorT;
```

```
    t: TEXT;
```

```
PROCEDURE learn_forward_model;
```

```
VAR i, iteration: INTEGER;
```

```
    outvector, flight, distance: vectorT;
```

```
    datavector: environ_vectorT;
```

```
    dim_forward_net: dimT;
```

```
    errortot, gamma_fwdmodel: REAL;
```

```
    f: TEXT;
```

```
BEGIN
```

```
    assign (f, 'fwdmodel.err');
```

```
    rewrite (f);
```

```
    dim_forward_net[input] := 2;
```

```
    dim_forward_net[hidden] := 5;
```

```
    dim_forward_net[output] := 1;
```

```
    init_net (forward_net, dim_forward_net);
```

```
    init_forward_model_data;
```

```
    gamma_fwdmodel := 1;
```

```
    iteration := 0;
```

```
    WHILE iteration < number_of_forward_iterations DO BEGIN
```

```
        inc (iteration);
```

```
        errortot := 0;
```

```

FOR i := 1 TO num_forward_vectors DO BEGIN
    fetch_vector (i, datavector);
    norm_vector (datavector);
    transform_vector (datavector, flight, distance);
    learn1 (flight, distance, forward_net, outvector, gamma_fwdmodel);
    errortot := errortot + sqr (distance[1]-outvector[1]);
END;
writeln (iteration, ' Fehler im forward-model: ',errortot:10:8);
writeln (f, errortot:10:8);
END;
write_net (forward_net, 'fwdmodel.net');
close (f);
END;

PROCEDURE test_forward_model;
VAR i: INTEGER;
    err: REAL;
    p: parameterT;
    outvector, flight, distance: vectorT;
    datavector: environ_vectorT;
    t: TEXT;
BEGIN
    init_forward_model_data;
    assign (t, 'fwdmodel.dat');
    rewrite (t);
    read_net (forward_net, 'fwdmodel.net');
    FOR i := 1 TO num_forward_vectors DO BEGIN
        fetch_vector (i, datavector);
        norm_vector (datavector);
        transform_vector (datavector, flight, distance);
        feedforward (forward_net, flight, outvector);
        datavector[sxnet] := outvector[1];
        write (t, (outvector[2]-0.1) * 5 + 7:4:3);
        unnorm_vector (datavector);
        datavector[sxmat] := compute_distance (datavector[vnull], datavector[alfa]);
        FOR p := vnull TO sxnet DO
            write (t, ' ',datavector[p]:4:3);
            writeln(t);
        END;
        close (t);
    END;
END;

PROCEDURE contextual_interference
    (block_size: INTEGER; VAR f: TEXT; forward_net: netT;
     VAR inverse_net: netT; gamma: REAL);
VAR iteration, itemnum, tasknum, i: INTEGER;
    datavector: environ_vectorT;
    l: parameterT;
BEGIN
    FOR l := vnull TO sxnet DO
        datavector[l] := 0;
    IF block_size = 0 {Zufällige auswahl} THEN BEGIN
        FOR itemnum := 1 TO number_of_inverse_iterations DO BEGIN
            datavector [sxmat] := width[random(4)+1];
            learn2 (f, datavector, forward_net, inverse_net, gamma);
        END;
    END;
END;

```



```

ELSE BEGIN
  iternum := 1;
  FOR iteration := 1 TO (number_of_inverse_iterations DIV block_size) DIV number_of_tasks DO BEGIN
    FOR tasknum := 1 TO number_of_tasks DO BEGIN
      FOR i := 1 TO block_size DO BEGIN
        datavector[sxmat] := width[tasknum];
        learn2 (f, datavector, forward_net, inverse_net, gamma);
        inc (iternum);
      END;
    END;
  END;
END;
END;
END;

PROCEDURE learn_inverse (vnet: STRING; gruppe: INTEGER);
VAR iteration, errarr_count, num_task: INTEGER;
    dim_inverse_net: dimT;
    vec: environ_vectorT;
    errortot, gamma_invmodel, gamma_null: REAL;
    flight, distance, outvector: vectorT;
    f: TEXT;
    n: STRING[1];
BEGIN
  read_net (forward_net, 'fwdmodel.net');
  dim_inverse_net[input] := 1;
  dim_inverse_net[hidden] := 5;
  dim_inverse_net[output] := 2;
  init_net (inverse_net, dim_inverse_net);
  assign (f, 'dfnt'+num_versuch+vnet+'.err');
  rewrite (f);
  errarr_count := 1;
  gamma_invmodel := 5;
  writeln ('Gruppe ' + gruppe);
  CASE gruppe OF
    1: contextual_interference (0,f,forward_net, inverse_net,gamma_invmodel);
        {block_size = 0 bedeutet Zufallsauswahl des Zielvektors}
    2: contextual_interference (1,f,forward_net, inverse_net,gamma_invmodel);
    3: contextual_interference (2,f,forward_net, inverse_net,gamma_invmodel);
    4: contextual_interference (5,f,forward_net, inverse_net,gamma_invmodel);
    5: contextual_interference (number_of_inverse_iterations DIV number_of_tasks,f,
        forward_net, inverse_net,gamma_invmodel);
  END;
  close (f);
END;

BEGIN {main}
writeln ('----- Neuer Versuch -----');
learn_forward_model;
test_forward_model;
assign (t, 'nets.tst');
rewrite (t);
FOR gruppe := 1 TO 5 DO BEGIN
  FOR neti := 1 TO 20 DO BEGIN
    str (gruppe:1,s1);
    str ((neti div 10):1,s2);
    str ((neti mod 10):1,s3);
    vnet := s1 + s2 + s3;

```

```
learn_inverse (vnet,gruppe);
write (t, num_versuch + vnet, '');
FOR i := 1 TO 7 DO BEGIN
    datavector [sxmat] := width[i];
    norm_vector (datavector);
    test_inverse_net (datavector, inverse_net, forward_net, test_vector);
    writeln ('vnull: ',test_vector[vnull]:6:4,' alfa: ',test_vector[alfa]:6:4,
            ' sx*:',test_vector [sxmat]:6:4,' sx: ',test_vector[sxnet]:6:4);
    write (t,test_vector[vnull]:6:4,' ',test_vector[alfa]:6:4,' ',
            test_vector [sxmat]:6:4,' ',test_vector[sxnet]:6:4,' ');
    END;
    writeln (t);
END;
END;
close (t);
END.
```

# **Personenregister**

## **A**

Adams ..... 22; 73  
 Anderson ..... 38; 81  
 Asatryan ..... 26

## **B**

Battig ..... 84  
 Baumann ..... 16  
 Beek ..... 18; 19; 57; 59  
 Bernstein ..... 13-19; 62  
 Bootsma ..... 57; 59  
 Bortz ..... 110; 112  
 Bullock ..... 52-63; 75; 78; 132  
 Bunz ..... 20

## **C**

Churchland ..... 132  
 Clark ..... 38  
 Creutzfeld ..... 35

## **D**

Deno ..... 58; 61  
 Descartes ..... 49  
 Dörner ..... 22; 133

## **F**

Feldman ..... 26; 31  
 Fitts ..... 56  
 Flash ..... 25; 70  
 Fodor ..... 34; 38; 50; 127  
 Furukawa ..... 78

## **G**

Gaudiano ..... 61; 62; 75  
 Georgopoulos ..... 56  
 Gibson ..... 18; 129  
 Gielen ..... 56  
 Grossberg ..... 52-63; 75; 78; 132

## **H**

Haas ..... 21  
 Haesner ..... 135  
 Haken ..... 19; 20; 59  
 Hall ..... 84  
 Hanson ..... 11  
 Hart ..... 30  
 Hebb ..... 37

Hegel ..... 49  
 Helm ..... 38  
 Heuer .. 6; 31; 58; 61; 126; 132; 133  
 Hinton ..... 34; 37; 40; 47; 52  
 Hirayama ..... 78  
 Hobbes ..... 49  
 Hogan ..... 25; 70  
 Holzkamp ..... 21; 32; 128  
 Hopfield ..... 41; 77; 79  
 Horak ..... 86; 87; 99; 112; 115  
 Hossner ..... 18; 23; 32; 133; 135  
 Hucho ..... 35

## **J**

James ..... 35; 36  
 Jennings ..... 63  
 Jensen ..... 30  
 Jordan ..... 52; 63-78; 88; 120-123  
 ..... 130; 132; 134

## **K**

Kalaska ..... 56  
 Kämmerling ..... 49  
 Kawato ..... 24; 26; 78  
 Keele ..... 22; 63  
 Keller ..... 126  
 Kelso ..... 19; 20; 29; 31; 57; 74  
 Kohonen ..... 38  
 Kugler ..... 19; 29  
 Kuhn ..... 38; 135  
 Künzell ..... 5; 6; 134  
 Kuperstein ..... 53

## **L**

Lames ..... 74  
 Lashley ..... 31; 37  
 Le Cun ..... 46  
 Lee ..... 19  
 Leibniz ..... 49  
 Lutz ..... 38

## **M**

Maeda ..... 78  
 Magill ..... 84  
 Marr ..... 14; 125  
 Marteniuk ..... 63  
 Massey ..... 56

Masson.....86; 87; 99; 112; 115  
 McClelland ..... 34; 37; 38; 125  
 McCulloch..... 36; 37; 41; 42  
 Meijer ..... 49  
 Memmi ..... 130  
 Minsky .....37; 38  
 Morasso .. 15; 53; 56; 57; 63; 76; 77  
 Morgan.....84; 101  
 Moxley ..... 83  
 Munzert ..... 18; 22; 128

*N*

Neumann ..... 21; 32; 127; 128; 135  
 Norman .....52; 130

*O*

Olivier ..... 18  
 Olson ..... 11

*P*

Papert .....37; 38  
 Parker ..... 46  
 Parks..... 127  
 Pitts ..... 36; 37; 41; 42  
 Prinz .....21; 127  
 Pylyshyn..... 34; 38; 127

*R*

Raab ..... 127  
 Ramacher..... 82  
 Rentsch ..... 135  
 Requin..... 12  
 Rojas .....7; 39; 40; 46; 47; 82  
 Rollfing ..... 135  
 Rosenbaum .....53; 56; 63; 65; 73  
 Rosenblatt.....37; 47  
 Rosenfeld.....38; 81  
 Roth ..... 5; 6; 22; 27; 49; 56; 127  
 Rumelhart .. 34; 37-40; 47; 52; 64-67  
 ..... 70-72; 88; 125; 132  
 Rutkowska ..... 128

*S*

Sanguinetti..... 53; 63; 76  
 Scheerer .....49; 50

Schlaak..... 135  
 Schmidt, G. .... 135  
 Schmidt, R.A. .... 13; 22; 27-30; 56  
 ..... 72-74; 83-86; 130; 134  
 Schmidt, R.C. .... 62  
 Schneider, K. .... 30  
 Schneider, W. .... 38  
 Schöner ..... 20; 74  
 Sejnowski..... 132  
 Shea..... 84; 101  
 Smolensky.....38; 39; 50; 125; 126  
 Spencer ..... 35  
 Stelmach..... 12  
 Sternad ..... 18; 19  
 Strube ..... 132  
 Suzuki ..... 26; 78

*T*

Thelen ..... 30  
 Thorndike..... 34; 36  
 Treffner ..... 62  
 Turvey .....19; 29; 62

*U*

Ulrich..... 30  
 Uno..... 26; 78

*V*

van Rossum..... 83; 84  
 van Wieringen..... 129  
 Vereijken..... 18  
 von Neumann ..... 11; 81

*W*

Walker ..... 35; 36  
 Wiemeyer..... 18; 22; 133  
 Wilberg ..... 59  
 Williams ..... 40; 47  
 Willimczik ..... 22

*Z*

Zanone ..... 74  
 Zelaznik ..... 56  
 Zernicke ..... 30